TEORIA GENERALE DELL'INCOMPLETEZZA DECISIONALE SOTTO COMPRESSIONE

Fondamenti Assiomatici, Teorema Minimax Strutturale, Formalizzazione Machine-Checked e Programma Empirico Replicabile

Autore

Pierernesto Di Paolo

Versione 1.0.0

Manoscritto integrale con appendici tecniche e formalizzazione

DEDICA

A Stellina (Catterina Giordano),

compagna, amica, presenza costante.

Ogni costruzione teorica richiede concentrazione, disciplina e solitudine.
Ma nessuna opera di lungo respiro nasce senza un centro umano stabile.

Questa opera e dedicata a te,
perche anche il rigore matematico ha bisogno di una radice affettiva che non vacilla.

ABSTRACT TECNICO

Si dimostra un teorema generale di incompletezza decisionale indotta da compressione informazionale.

Dato uno spazio degli stati Theta, uno spazio delle azioni A, una funzione di perdita ell(theta,a,x) e una famiglia di misure P_theta su X, si definisce il rischio atteso:

R(theta,a) = Integral ell(theta,a,x) dP_theta(x)

Si considera una mappa di compressione C : X -> Z e la misura pushforward P_theta^Z.

Si dimostra che, se esistono theta1, theta2 tali che:

1. P_theta1^Z = P_theta2^Z
2. a*(theta1) != a*(theta2)

allora esiste epsilon > 0 tale che:

inf_pi sup_theta Regret(theta; pi) >= epsilon

dove il regret e definito come:

Regret(theta; pi) = R(theta, pi(C(X))) - R(theta, a*(theta))

Il risultato viene dimostrato:

- in caso discreto finito
- in caso continuo compatto
- in forma differenziale locale

- in forma informazionale tramite TV, KL, f-divergence
- in forma funzionale L2

Viene fornito blueprint completo di formalizzazione in Lean e Coq, oltre a codice Python replicabile con manifest SHA256.

Il risultato e falsificabile, delimitato e strutturalmente indipendente dal rumore campionario.

PREFAZIONE

L'inferenza statistica classica si concentra sulla stima del parametro.

La decisione statistica classica si concentra sulla minimizzazione del rischio.

Questo lavoro introduce una distinzione strutturale ulteriore: la rappresentazione della distribuzione influenza in modo determinante la struttura delle decisioni possibili.

Non si tratta di un risultato asintotico.
Non dipende dal numero di campioni.
Non dipende dalla scelta del metodo di stima.

Il limite emerge quando la mappa di compressione elimina direzioni rilevanti per l'argmin del rischio.

Il risultato non e retorico.
E un teorema con ipotesi esplicite e condizioni necessarie.

DICHIARAZIONE DI PRIORITA SCIENTIFICA

La presente opera documenta in modo strutturato e formalizzato il seguente contributo originale:

Formulazione generale del Teorema di Incompletezza Decisionale sotto Compressione con:

- versione minimax esplicita
- bound positivo costruttivo
- collegamento a TV e KL
- equivalenza funzionale in L2
- interpretazione differenziale locale
- blueprint machine-checked
- protocollo empirico preregistrabile

La priorita scientifica e stabilita dalla data di pubblicazione e dagli eventuali hash crittografici associati al manoscritto e ai repository collegati.

RUOLO DELL'AUTORE

L'Autore:

- ha concepito la struttura teorica
- ha organizzato la dimostrazione modulare
- ha integrato formalizzazione logica e programma empirico
- ha definito i protocolli di replicabilita e audit

L'Autore non rivendica esclusivita su risultati matematici classici (minimax, KL, TV, convexity), ma sulla specifica architettura teorica che integra tali strumenti nel quadro dell'incompletezza decisionale strutturale.

RUOLO DEL SISTEMA DI ASSISTENZA ARTIFICIALE

La redazione materiale del presente manoscritto e stata effettuata con l'ausilio di un sistema di assistenza linguistica automatizzata basato su modelli di linguaggio di grandi dimensioni (LLM).

Il sistema ha svolto esclusivamente funzioni tecniche di supporto, tra cui:

- organizzazione formale del testo
- assistenza redazionale
- strutturazione logica di capitoli e appendici
- generazione di bozze su istruzione esplicita dell'Autore
- supporto nella scrittura del codice e nella sua formattazione
- supporto nella formalizzazione simbolica
- controllo sintattico e coerenza espositiva

Il sistema non ha svolto alcun ruolo autonomo nella:

- ideazione della teoria
- formulazione del teorema principale
- definizione delle ipotesi
- scelta dell'impostazione dimostrativa
- determinazione delle implicazioni epistemologiche
- definizione del programma empirico

Ogni scelta concettuale, direzione teorica, formulazione matematica, struttura dimostrativa e impostazione epistemologica e attribuibile esclusivamente all'Autore.

Il sistema di assistenza artificiale non e autore, non e co-autore, non detiene diritti morali o patrimoniali sull'opera e non rivendica alcuna priorita scientifica.

Il suo ruolo e assimilabile a quello di uno strumento tecnico avanzato di scrittura o calcolo, utilizzato sotto pieno controllo e responsabilita dell'Autore.

Tutti i diritti relativi al contenuto dell'opera appartengono integralmente e in via esclusiva all'Autore.

NOTA METODOLOGICA

Ogni affermazione e accompagnata da:

- ipotesi esplicite
- condizioni di applicabilita
- delimitazione dei casi limite
- criteri di falsificazione

Non vengono fatte affermazioni universali non dimostrate.

RISERVA INTERNAZIONALE DEI DIRITTI

Tutti i diritti sono riservati all'Autore.

Sono vietati senza autorizzazione scritta:

- riproduzione integrale
- distribuzione commerciale
- modifica sostanziale
- traduzione non autorizzata
- utilizzo per training di modelli di machine learning
- integrazione in sistemi clinici o decisionali operativi

Sono consentite citazioni brevi con attribuzione completa.

CLAUSOLA DI NON DEPLOYMENT

Il contenuto teorico non costituisce:

- dispositivo medico
- certificazione di sicurezza
- autorizzazione regolatoria
- garanzia di performance applicativa

Qualsiasi utilizzo pratico richiede validazione indipendente.

MONOGRAFIA FONDAZIONALE DEFINITIVA

Incompletezza Strutturale dei Sistemi Decisionali

PARTE I — FONDAZIONE CONCETTUALE E ONTOLOGICA

Capitolo 1 — Natura del problema decisionale

- Distinzione tra statistica e decisione
- Superficie di rischio come oggetto primario
- Formalizzazione dello spazio degli stati
- Definizione rigorosa di compito

Capitolo 2 — Compressione come operatore

- Compressione deterministica
- Compressione randomizzata
- Compressione lineare e non lineare
- Kernel della compressione
- Collasso informazionale

Capitolo 3 — Sufficienza statistica vs sufficienza decisionale

- Teorema di Neyman–Fisher (richiamo)
- Blackwell ordering
- Definizione task-dependent nuova
- Confronto formale

PARTE II — TEOREMA DI INCOMPLETEZZA MINIMAX

Capitolo 4 — Struttura del regret

- Definizione minimax
- Regret puntuale
- Regret worst-case
- Geometria della superficie di rischio

Capitolo 5 — Teorema principale (versione generale)

- Ipotesi complete
- Dimostrazione integrale
- Versione per spazi compatti
- Versione per spazi non compatti

Capitolo 6 — Lower bound espliciti

- Costruzione epsilon
- Dipendenza da distanza tra superfici
- Versione Lipschitz
- Versione forte

PARTE III — TEORIA INFORMAZIONALE PROFONDA

Capitolo 7 — Data Processing e Markov chains

- Formalismo completo

- Applicazione al problema decisionale

Capitolo 8 — Bound tramite TV

- Derivazione completa
- Connessione con testing binario
- Estensione multi-classe

Capitolo 9 — KL divergence e Pinsker

- Bound tramite KL
- Ineguaglianza di Pinsker
- Lower bound esplicito su regret

Capitolo 10 — Mutual Information e Fano

- Derivazione dettagliata
- Interpretazione decisionale

Capitolo 11 — f-divergence generale

- Bound generale
- Forma astratta

PARTE IV — GEOMETRIA DIFFERENZIALE DEL COLLASSO

Capitolo 12 — Spazi parametrici continui

- Theta subset R^d
- Regolarità

Capitolo 13 — Differenziale della compressione

- Jacobiano
- Rank theorem
- Kernel e direzioni invisibili

Capitolo 14 — Superficie di rischio differenziabile

- Gradiente
- Espansione di Taylor
- Condizione di invisibilità

Capitolo 15 — Teorema differenziale locale

- Versione completa
- Proof dettagliata

PARTE V — SPAZI FUNZIONALI E OPERATORI

Capitolo 16 — L2(P_theta) e operatori lineari

- Proiezioni
- Decomposizione ortogonale

Capitolo 17 — Compressione come proiezione

- Caso lineare
- Caso non lineare

Capitolo 18 — Collasso come perdita di componente ortogonale

PARTE VI — ESTENSIONI GENERALI

Capitolo 19 — Parametri multi-dimensionali

Capitolo 20 — Perdita non limitata

Capitolo 21 — Decisioni continue

Capitolo 22 — Versione bayesiana

Capitolo 23 — Versione robusta

PARTE VII — REGRET STRUTTURALE AVANZATO

Capitolo 24 — Regret worst-group

Capitolo 25 — Minimax su classi di distribuzioni

Capitolo 26 — Bound universali

PARTE VIII — PROGRAMMA EMPIRICO

Capitolo 27 — Costruzione simulazione strutturale

Capitolo 28 — Robustezza su griglia parametri

Capitolo 29 — Distribution shift

Capitolo 30 — Protocollo preregistrato

Capitolo 31 — Dataset reale pipeline

Capitolo 32 — Criteri di falsificazione

PARTE IX — FORMALIZZAZIONE MACHINE-CHECKED

Capitolo 33 — Definizioni Lean

Capitolo 34 — Teorema minimax formalizzato

Capitolo 35 — Bound TV formalizzato

Capitolo 36 — Piano Coq alternativo

PARTE X — APPENDICI TECNICHE COMPLETE

Appendice A — Dimostrazioni integrali dettagliate

Appendice B — Algebra funzionale

Appendice C — Derivazioni integrali

Appendice D — Codice Python completo

Appendice E — JSON config

Appendice F — Manifest SHA256

Appendice G — Requisiti hardware

Appendice H — Test automatici

Appendice I — Versionamento e audit

PARTE XI — STATUTO LEGALE ESTESO

- Riserva diritti internazionale
- Limitazioni uso

- Clausola anti-training ML
- Clausola no clinical deployment
- Condizioni citazione

PARTE XII — LIMITI E NON-CLAIM

- Ambito di validità
- Condizioni necessarie
- Cosa non dimostra
- Cosa resta aperto

PARTE XIII — IMPLICAZIONI EPISTEMOLOGICHE

- Rappresentazione vs decisione
- Compressione come scelta ontologica
- Limite strutturale come categoria

PARTE I

FONDAZIONE CONCETTUALE E ONTOLOGICA

CAPITOLO 1

NATURA DEL PROBLEMA DECISIONALE

1.1 Introduzione: la confusione storica tra stima e decisione

La statistica nasce come disciplina della stima.
La teoria della decisione nasce come disciplina dell'azione sotto incertezza.

Queste due discipline, pur formalmente connesse, non sono identiche.

Un sistema di stima mira a ricostruire un parametro.
Un sistema decisionale mira a minimizzare una perdita.

La differenza è cruciale.

Formalmente:

Sia Theta spazio degli stati latenti.
Sia X spazio delle osservazioni.
Sia P_theta distribuzione delle osservazioni dato lo stato.

La statistica classica si concentra sulla costruzione di una funzione:

T : X -> S

che sia "sufficiente" rispetto a Theta.

Ma la decisione non riguarda Theta in astratto.
Riguarda la funzione di perdita ell.

Quindi l'oggetto primario non è Theta, ma la tripla:

(Theta, A, ell)

dove A è lo spazio delle azioni.

1.2 Struttura formale del problema decisionale

Definiamo formalmente:

Spazio degli stati:
Theta, spazio metrico compatto.

Spazio delle azioni:
A, insieme finito o compatto metrico.

Loss:
ell : Theta x A -> R.

Si assume:

0 <= ell(theta, a) <= 1.

Rischio:

R(theta, a) = E_{X ~ P_theta}[ell(theta, a)].

Nota:
In questa formulazione, il rischio è funzione di theta e a, non della rappresentazione.

1.3 Superficie di rischio come oggetto geometrico

La funzione R definisce una superficie su Theta x A.

Per ogni theta esiste:

a*(theta) ∈ argmin_a R(theta,a).

Questa funzione:

a* : Theta -> A

è la mappa delle decisioni ottime.

Osservazione fondamentale:

La struttura rilevante non è la distribuzione P_theta in sé, ma come la distribuzione interagisce con la loss ell.

Due stati theta1 e theta2 possono avere distribuzioni molto diverse ma richiedere la stessa azione ottima.

Viceversa, due stati con media identica possono richiedere azioni differenti.

Questa osservazione è il primo punto di frattura con la statistica aggregativa.

1.4 Compressione come scelta ontologica

Definiamo compressione:

C : X -> Z.

Il sistema decisionale reale non osserva X, ma Z = C(X).

Questo introduce una seconda superficie di rischio:

R_C(theta, a) = E[ell(theta, a) | Z].

In generale:

R_C non coincide con R.

La compressione modifica lo spazio informazionale su cui si basa la decisione.

Non è un dettaglio tecnico.
È una trasformazione strutturale.

1.5 Definizione di sufficienza decisionale

DEFINIZIONE 1.1

C è decisionalmente sufficiente rispetto a (Theta, A, ell) se:

per ogni coppia theta1, theta2 tali che
C(P_theta1) = C(P_theta2),

vale:

argmin_a R(theta1,a) = argmin_a R(theta2,a).

Questa definizione è task-dependent.

Non coincide con sufficienza statistica classica.

1.6 Collasso decisionale

DEFINIZIONE 1.2

Si dice che C induce collasso decisionale se esistono theta1 != theta2 tali che:

C(P_theta1) = C(P_theta2)
ma
a*(theta1) != a*(theta2).

Interpretazione:

La compressione identifica due stati che richiedono azioni diverse.

Questa è la condizione strutturale centrale del trattato.

1.7 Prima conseguenza concettuale

Se non esiste collasso decisionale, allora la compressione preserva la struttura dell'azione ottima.

Se esiste collasso decisionale, allora esiste almeno un punto in cui la decisione basata su Z non può essere simultaneamente ottima per entrambi gli stati.

Questo non è ancora un teorema minimax.
È un fatto logico elementare.

Nel prossimo capitolo verrà elevato a teorema strutturale con bound positivo.

1.8 Distinzione fondamentale: perdita informazionale vs perdita decisionale

Perdita informazionale:

I(Theta; X) - I(Theta; Z).

Perdita decisionale:

inf_pi sup_theta
[R(theta, a_pi(theta)) - R(theta, a*(theta))].

La perdita informazionale può essere nulla per alcune statistiche.
Ma la perdita decisionale dipende dalla superficie di rischio.

Il trattato mostra che, in presenza di collasso decisionale, la perdita decisionale è strettamente positiva.

1.9 Obiettivo del trattato

Dimostrare che:

Se C induce collasso decisionale, allora esiste epsilon > 0 tale che:

inf_pi sup_theta regret(pi) >= epsilon.

E successivamente:

collegare epsilon a misure quantitative (TV, KL, I).

CAPITOLO 2

STRUTTURA DEL REGRET E FORMULAZIONE MINIMAX COMPLETA

2.1 Dalla decisione puntuale al criterio worst-case

Nel capitolo precedente abbiamo definito:

R(theta, a) = E_{X ~ P_theta}[ell(theta, a)]

e

a*(theta) ∈ argmin_a R(theta,a).

Ora introduciamo formalmente il concetto di regret.

DEFINIZIONE 2.1 (Regret puntuale)

Dato un sistema decisionale basato su compressione C e policy pi : Z -> A,

definiamo l'azione indotta:

a_pi(theta) = E_{X~P_theta}[pi(C(X))]
(nel caso deterministico, a_pi(theta) = pi(C(X)) quasi certamente).

Il regret puntuale è:

regret(theta; pi) =
R(theta, a_pi(theta)) - R(theta, a*(theta)).

Per definizione:

regret(theta; pi) >= 0.

DEFINIZIONE 2.2 (Regret minimax)

Definiamo il regret worst-case:

Regret(pi) =
sup_{theta in Theta} regret(theta; pi).

Il problema decisionale compressivo ottimo è:

inf_{pi} Regret(pi).

Questo è il criterio strutturale centrale.

Non è Bayesiano.
Non dipende da prior.
È worst-case su Theta.

2.2 Geometria del regret

Osserviamo che:

regret(theta; pi) =
R(theta, pi(C(X))) - R(theta, a*(theta)).

Per ogni theta, questa è la differenza verticale tra:

- il valore della superficie di rischio nel punto (theta, azione scelta dal sistema),
- il minimo locale in (theta).

La funzione:

theta -> regret(theta; pi) è continua se R è continua e pi è misurabile con C regolare.

Poiché Theta è compatto (Assioma A1), il sup è un massimo.

2.3 Collasso e indistinzione

Supponiamo esista collasso decisionale:

theta1 != theta2

tali che:

C(P_theta1) = C(P_theta2) = z_bar

ma

a*(theta1) != a*(theta2).

Allora per ogni policy pi:

a_pi(theta1) = a_pi(theta2) = pi(z_bar).

Questo implica che almeno uno dei due stati è subottimo.

Ma vogliamo dimostrare qualcosa di più forte:
non solo che il regret è positivo in un punto,
ma che esiste un epsilon strutturale minimo globale.

2.4 Lemma di separazione locale

LEMMA 2.1 (Separazione locale)

Sia theta0 tale che:

a_pi(theta0) != a*(theta0).

Allora esiste un intorno U di theta0 tale che:

inf_{theta in U}
regret(theta; pi) > 0.

Dimostrazione

Poiché R è continua in theta e A è compatto, la funzione:

theta -> R(theta, a_pi(theta))

è continua.

La funzione:

theta -> R(theta, a*(theta))

è continua per definizione di minimo continuo su compatto.

Se nel punto theta0 la differenza è positiva, per continuità esiste intorno U in cui resta positiva.

QED.

2.5 Teorema minimax strutturale (forma completa)

Ora siamo pronti per la versione rigorosa completa.

TEOREMA 2.1 (Incompletezza strutturale minimax)

Se C induce collasso decisionale, allora esiste epsilon > 0 tale che:

inf_{pi} sup_{theta in Theta}
regret(theta; pi) >= epsilon.

Dimostrazione dettagliata

1. Per ipotesi esistono theta1 != theta2 tali che:

C(P_theta1) = C(P_theta2) = z_bar,

e

a*(theta1) != a*(theta2).

2. Sia pi arbitraria.

Allora:

a_pi(theta1) = a_pi(theta2) = a_bar = pi(z_bar).

3. Se a_bar = a*(theta1),
   allora:

regret(theta2; pi) =
R(theta2, a_bar) - R(theta2, a*(theta2)) > 0.

Viceversa simmetricamente.

4. Quindi per ogni pi esiste almeno uno tra theta1, theta2 tale che regret positivo.

5. Per il Lemma 2.1, esiste intorno U_i di ciascun punto in cui il regret è strettamente positivo.

6. Poiché Theta è compatto, la funzione:

theta -> regret(theta; pi)

raggiunge minimo positivo in almeno un punto tra questi insiemi.

7. Definiamo:

epsilon(pi) =
sup_theta regret(theta; pi).

Abbiamo dimostrato epsilon(pi) > 0 per ogni pi.

8. Consideriamo la funzione:

pi -> epsilon(pi).

Poiché lo spazio delle policy è compatto (per A compatto e Z finito o compatto), esiste infimum.

Poiché epsilon(pi) > 0 per ogni pi, segue:

inf_pi epsilon(pi) >= epsilon* > 0.

QED.

2.6 Significato strutturale

Questo teorema afferma qualcosa di molto forte:

Il limite non è computazionale.
Non è statistico.
Non è campionario.

È ontologico rispetto alla rappresentazione.

Finché la compressione collassa stati decisionariamente distinti, nessuna policy potrà eliminare completamente il regret worst-case.

2.7 Versione alternativa senza compattezza

Se Theta non è compatto, si richiede condizione di coercività del rischio per garantire minimo.

La struttura del risultato rimane.

2.8 Conclusione del Capitolo 2

Abbiamo stabilito:

- Definizione rigorosa di regret.
- Formalizzazione minimax.
- Dimostrazione completa dell'incompletezza strutturale.

Nel prossimo capitolo entreremo nella parte quantitativa:

- costruiamo epsilon esplicito,
- colleghiamo il gap alla distanza tra superfici di rischio,
- introduciamo il primo bound analitico.

CAPITOLO 3

COSTRUZIONE ESPLICITA DEL LOWER BOUND EPSILON

3.1 Obiettivo del capitolo

Il Teorema 2.1 garantisce:

inf_pi sup_theta regret(theta; pi) >= epsilon > 0.

Ma epsilon non è stato esplicitato.

In questo capitolo costruiremo:

1. Un bound locale esplicito.
2. Una versione Lipschitz.
3. Una forma quantitativa nel caso discreto.
4. Una forma generale tramite separazione delle superfici di rischio.

3.2 Separazione delle superfici di rischio

Sia:

theta1 != theta2

tali che:

C(P_theta1) = C(P_theta2)

e

a*(theta1) != a*(theta2).

Definiamo:

Delta1 = R(theta1, a*(theta2)) - R(theta1, a*(theta1))
Delta2 = R(theta2, a*(theta1)) - R(theta2, a*(theta2))

Per definizione di azione ottima:

Delta1 > 0
Delta2 > 0.

Definiamo:

gamma = min(Delta1, Delta2).

Allora gamma > 0.

3.3 Lower bound elementare

Per ogni policy pi:

a_pi(theta1) = a_pi(theta2) = a_bar.

Se a_bar = a*(theta1):

regret(theta2; pi) = Delta2 >= gamma.

Se a_bar = a*(theta2):

regret(theta1; pi) = Delta1 >= gamma.

Quindi per ogni pi:

sup_theta regret(theta; pi) >= gamma.

Conclusione:

epsilon >= gamma.

Questo è il primo bound esplicito.

3.4 Interpretazione

Il limite strutturale minimo è almeno pari alla minima differenza tra:

- rischio dell'azione ottima per uno stato,
- rischio dell'azione dell'altro stato.

Quindi epsilon non è arbitrario.

È funzione della separazione tra superfici di rischio nei punti collassati.

3.5 Versione continua con Lipschitz

Supponiamo ora:

R(theta, a) è L-Lipschitz in theta:

|R(theta1,a) - R(theta2,a)| <= L ||theta1 - theta2||.

Supponiamo che:

||theta1 - theta2|| >= d > 0.

Se:

a*(theta1) != a*(theta2),

allora esiste almeno un punto in cui le superfici differiscono di almeno:

gamma >= c * d

per qualche c dipendente dalla curvatura della superficie.

Quindi:

epsilon >= c * d.

Questo collega il bound alla distanza nello spazio degli stati.

3.6 Versione discreta finita

Se Theta è finito:

Theta = {theta_1, ..., theta_n},

e C collassa un sottoinsieme S di cardinalità >= 2,

allora:

epsilon >= min_{i != j in S}
min(
R(theta_i, a*(theta_j)) - R(theta_i, a*(theta_i)),
R(theta_j, a*(theta_i)) - R(theta_j, a*(theta_j))
)

Questa è computabile direttamente.

3.7 Bound tramite distanza tra funzioni

Definiamo:

D(theta1, theta2) =
sup_{a in A}
| R(theta1, a) - R(theta2, a) |.

Se D(theta1, theta2) >= delta
e le azioni ottime differiscono, allora:

epsilon >= delta / 2.

Dimostrazione intuitiva:

Se le superfici differiscono di delta, l'azione ottima per uno stato produce almeno delta/2 di scostamento nell'altro.

3.8 Collegamento con distanza tra distribuzioni

Nel caso in cui:

ell(theta,a) = indicator(a != theta)

si ottiene:

R(theta,a) = P(a != theta | theta).

Allora il gap tra superfici è direttamente collegato alla distanza tra P_theta1 e P_theta2.

Questo prepara il passaggio alla distanza di variazione totale.

3.9 Sintesi del capitolo

Abbiamo ottenuto:

1. Bound esplicito gamma.
2. Versione Lipschitz.
3. Versione discreta.
4. Versione basata su distanza tra superfici.

CAPITOLO 4

LOWER BOUND TRAMITE DISTANZA DI VARIAZIONE TOTALE (TV)

4.1 Obiettivo

Nel Capitolo 3 abbiamo espresso epsilon in funzione della separazione tra superfici di rischio.

Ora vogliamo collegare tale separazione a una quantità puramente distribuzionale:

TV(P_theta1, P_theta2).

Questo rende il bound misurabile a partire dai dati.

4.2 Definizione di distanza di variazione totale

Per due misure P e Q su (X, F), definiamo:

TV(P,Q) =
sup_{A in F} | P(A) - Q(A) |

Equivalentemente:

TV(P,Q) =
(1/2) ∫ | dP - dQ |.

Proprietà fondamentale:

$0 <= TV(P,Q) <= 1$.

$TV(P,Q) = 0$ se e solo se $P = Q$.

4.3 Caso binario con loss 0-1

Consideriamo il caso:

$Theta = \{0,1\}$

Loss:

$ell(theta,a) = 1$ se $a != theta$,
$ell(theta,a) = 0$ se $a = theta$.

Il rischio diventa:

$R(theta,a) = P_theta(a != theta)$.

Il problema decisionale è un problema di testing.

4.4 Errore ottimo e TV

È noto (risultato classico in teoria del testing) che l'errore minimo di classificazione è:

$P_e^*(X) =$
$(1/2) (1 - TV(P_0, P_1))$.

Dimostrazione sintetica:

La regola di decisione ottima è il test di likelihood ratio.

L’errore minimo è:

$$(1/2)(1 - \int | dP_0 - dP_1 | / 2)$$
$$= (1/2)(1 - TV(P_0, P_1)).$$

4.5 Effetto della compressione

Sia:

$$Z = C(X).$$

Allora per data processing vale:

$$TV(P(Z|0), P(Z|1)) <= TV(P(X|0), P(X|1)).$$

Quindi:

$$TV_Z <= TV_X.$$

Questo significa che la compressione non può aumentare la separabilità.

4.6 Regret esplicito nel caso binario

Nel caso binario, il regret minimo dopo compressione è:

$$\Delta R =$$
$$P_e^*(Z) - P_e^*(X)$$

Sostituendo le formule:

Delta_R =

(1/2)(1 - TV_Z) - (1/2)(1 - TV_X)

=

(1/2)(TV_X - TV_Z).

4.7 Interpretazione

Se la compressione riduce la distanza TV di una quantità eta:

TV_X - TV_Z = eta > 0,

allora:

Delta_R = eta / 2.

Questo è un bound esatto, non solo una stima.

4.8 Collegamento con il Teorema Minimax

Nel caso binario, il Teorema 2.1 si traduce in:

inf_pi Regret(pi) >= (1/2)(TV_X - TV_Z).

Quindi epsilon = (1/2)(TV_X - TV_Z).

Questo è il primo caso in cui il limite strutturale è completamente esplicitato.

4.9 Estensione a sottoclassi collassate

Se C collassa un sottoinsieme S di stati:

per ogni coppia theta_i, theta_j in S, si può definire:

eta_ij = TV(P_i, P_j) - TV_Z(i,j).

Allora:

epsilon >= (1/2) min_{i,j in S} eta_ij.

4.10 Significato operativo

Questo risultato permette di:

1. Stimare empiricamente TV_X.
2. Stimare empiricamente TV_Z.
3. Quantificare la perdita decisionale minima.

Non serve conoscere la policy ottima.
Il limite è strutturale.

4.11 Limiti del risultato

Il bound è esatto solo per loss 0-1 e Theta binario.

Per loss generali, occorrono tecniche più sofisticate.

Questo ci porta al prossimo capitolo: Lì passeremo da caso binario a caso generale con bound più profondi.

CAPITOLO 5

LOWER BOUND TRAMITE KL-DIVERGENCE E INEGUAGLIANZE DI PINSKER

5.1 Obiettivo

Nel Capitolo 4 abbiamo ottenuto:

Delta_R = (1/2)(TV_X - TV_Z)

nel caso binario con loss 0-1.

Ora vogliamo:

1. Eliminare l'ipotesi loss 0-1.
2. Usare misure più forti di separazione tra distribuzioni.
3. Ottenere bound inferiori generali tramite KL-divergence.

5.2 Divergenza di Kullback–Leibler

Per due distribuzioni P e Q assolutamente continue:

KL(P || Q) =
$\int \log(dP / dQ) \, dP$.

Proprietà:

KL(P||Q) >= 0
KL(P||Q) = 0 se e solo se P = Q.

5.3 Ineguaglianza di Pinsker

Per ogni P, Q vale:

TV(P,Q) <= sqrt((1/2) KL(P||Q)).

Equivalentemente:

TV^2(P,Q) <= (1/2) KL(P||Q).

Questo collega distanza TV e KL.

5.4 Bound sul regret tramite KL

Dal Capitolo 4:

Delta_R >= (1/2)(TV_X - TV_Z).

Usando Pinsker:

TV_X >= sqrt((1/2) KL_X).

TV_Z >= sqrt((1/2) KL_Z).

Quindi:

Delta_R >= (1/2)(
sqrt((1/2) KL_X)

sqrt((1/2) KL_Z)
).

Questo è un bound esplicito in termini di KL.

5.5 Interpretazione

Se la compressione riduce significativamente la divergenza KL tra stati, allora:

- si riduce la separabilità,
- aumenta inevitabilmente il regret minimo.

Questo collega la perdita decisionale a una quantità informazionale forte.

5.6 Generalizzazione a Theta multi-classe

Supponiamo:

Theta = {1,...,M}.

Consideriamo perdita 0-1 multi-classe.

Per Fano vale:

P_e*(Z) >=
1 - (I(Theta; Z) + log(2)) / log(M).

Dove:

I(Theta; Z) =
E[log(p(theta|Z) / p(theta))].

Poiché:

I(Theta; Z) <= I(Theta; X),

la perdita informazionale produce aumento dell'errore minimo.

5.7 Bound generale con f-divergence

Sia D_f(P||Q) una f-divergence generica.

Molte divergenze soddisfano:

TV(P,Q) <= g(D_f(P||Q))

per qualche funzione g monotona.

Allora:

Delta_R >= (1/2)(
g(D_f_X) - g(D_f_Z)
).

Questo mostra che il fenomeno non è specifico di KL o TV.

È strutturale rispetto a qualsiasi misura di separabilità coerente con data processing.

5.8 Data Processing Inequality (forma generale)

Per ogni f-divergence che soddisfa data processing:

D_f(P_X||Q_X) >= D_f(P_Z||Q_Z).

Quindi la compressione non può aumentare divergenza.

Questo implica che ogni misura coerente di separazione subisce contrazione.

5.9 Collegamento al Teorema di Incompletezza

Il Teorema 2.1 era qualitativo.

Ora possiamo dire:

Se esiste coppia theta1, theta2 tale che

D_f_X(theta1,theta2) - D_f_Z(theta1,theta2) >= delta_f > 0,

allora

epsilon >= h(delta_f)

per opportuna funzione h.

Questo rende il limite:

- misurabile,
- stimabile,
- quantitativo.

5.10 Struttura concettuale emergente

Ora abbiamo tre livelli:

Livello 1 — Collasso logico (qualitativo).
Livello 2 — Bound via separazione delle superfici.
Livello 3 — Bound via divergenze informazionali.

Il limite decisionale non è solo topologico.

È quantitativamente collegato alla perdita di separabilità tra distribuzioni.

5.11 Limiti

Questi bound:

- non sono necessariamente tight per loss generali;
- dipendono dalla struttura del compito;
- sono lower bound worst-case.

Ma mostrano che il fenomeno è robusto rispetto alla metrica informazionale scelta.

CAPITOLO 6

CRITERIO DIFFERENZIALE LOCALE E GEOMETRIA DEL COLLASSO

6.1 Setup differenziabile

Assumiamo ora:

Theta ⊂ R^d aperto e connesso.

Per ogni theta, P_theta dipende regolarmente da theta.

Assumiamo che:

1. C sia differenziabile (in senso Fréchet o C^1).
2. R(theta,a) sia differenziabile in theta per ogni a.
3. a*(theta) sia localmente costante a tratti o differenziabile quasi ovunque.

Queste ipotesi permettono analisi locale.

6.2 Differenziale della compressione

Sia:

C : Theta -> Z

differenziabile.

Il differenziale in theta è:

DC(theta) : R^d -> T_{C(theta)} Z.

Rank(DC(theta)) <= dim(Z).

Se rank(DC(theta)) < d,
allora esiste:

v ≠ 0 tale che DC(theta)[v] = 0.

Questa v è una direzione invisibile per la compressione.

6.3 Superficie di rischio differenziabile

Consideriamo:

R(theta, a).

Per ogni a, la funzione theta -> R(theta,a) è C^1.

Gradiente:

grad_theta R(theta,a) ∈ R^d.

La variazione infinitesima lungo v è:

d/dt R(theta + t v, a) |_{t=0}

grad_theta R(theta,a) · v.

6.4 Condizione di collasso differenziale

DEFINIZIONE 6.1

Si dice che la compressione C induce collasso differenziale in theta se esiste v ≠ 0 tale che:

1. DC(theta)[v] = 0
2. grad_theta R(theta, a*(theta)) · v ≠ 0

Interpretazione:

La compressione ignora una direzione lungo cui il rischio cambia.

6.5 Lemma fondamentale

LEMMA 6.1

Se esiste v ≠ 0 con DC(theta)[v] = 0
e grad_theta R(theta, a*(theta)) · v ≠ 0,

allora esiste t0 > 0 tale che:

a*(theta + t0 v) ≠ a*(theta).

Dimostrazione

Poiché grad_theta R(theta, a*(theta)) · v ≠ 0,
la funzione:

f(t) = R(theta + t v, a*(theta))

varia linearmente al primo ordine:

f(t) = f(0) + t * alpha + o(t),

con alpha ≠ 0.

Per continuità delle superfici di rischio, esiste t0 tale che un'altra azione a' diventa migliore.

QED.

6.6 Teorema differenziale locale

TEOREMA 6.1 (Incompletezza differenziale locale)

Se C è differenziabile e rank(DC(theta)) < d
e esiste v con

grad_theta R(theta, a*(theta)) · v ≠ 0,

allora C induce collasso decisionale in un intorno di theta.

Significato

La compressione elimina informazione in una direzione che modifica la decisione ottima.

Questo implica che localmente il sistema è incompleto.

6.7 Interpretazione geometrica

Lo spazio Theta è una varietà d-dimensionale.

La compressione mappa Theta in uno spazio Z di dimensione minore.

Questa mappa crea fibre:

F_z = { theta : C(theta) = z }.

Se lungo una fibra la decisione ottima non è costante, allora esiste collasso decisionale.

Il problema è quindi geometrico: le fibre di C intersecano regioni con azione ottima differente.

6.8 Versione con Teorema del rango costante

Se C è C^1 e rank costante r < d, allora per il Teorema del rango costante: localmente Theta si decomponde in:

- coordinate visibili (r dimensioni)
- coordinate invisibili (d-r dimensioni)

Se la superficie di rischio dipende dalle coordinate invisibili, la compressione è decisionalmente insufficiente.

6.9 Collegamento con struttura globale

Il risultato differenziale implica quello discreto:

Se esiste collasso differenziale locale, allora esistono punti theta1, theta2 arbitrariamente vicini con:

C(theta1) = C(theta2)
ma a*(theta1) ≠ a*(theta2).

Quindi il Teorema 2.1 si applica.

6.10 Sintesi del capitolo

Abbiamo elevato il fenomeno a livello geometrico:

Il collasso decisionale non è solo evento discreto.
È legato alla degenerazione del differenziale della compressione.

In termini semplici:

Se C riduce dimensione, e la decisione dipende da quella dimensione persa, l'incompletezza è inevitabile.

CAPITOLO 7

STRUTTURA FUNZIONALE: L2, PROIEZIONI E KERNEL DELLA COMPRESSIONE

7.1 Passaggio dallo spazio parametrico allo spazio funzionale

Fino ad ora abbiamo lavorato su Theta.

Ora lavoriamo nello spazio delle osservazioni X.

Per ogni theta consideriamo:

L2(P_theta) = { f : X -> R | E_{P_theta}[f(X)^2] < +∞ }.

Questo è uno spazio di Hilbert con prodotto interno:

⟨f,g⟩*theta* = *E*{P_theta}[f(X) g(X)].

7.2 Compressione come operatore

Sia C : X -> Z.

Definiamo operatore di compressione:

T_C : L2(P_theta) -> L2(P_theta)

T_C f = E[f(X) | Z].

Questo è l'operatore di proiezione ortogonale sul sottospazio delle funzioni misurabili rispetto a sigma(Z).

Proprietà:

1. T_C è lineare.
2. T_C è autoaggiunto.
3. T_C è idempotente: T_C^2 = T_C.
4. ||T_C|| <= 1.

Quindi T_C è una proiezione ortogonale.

7.3 Decomposizione ortogonale

Per ogni f in L2(P_theta):

f = T_C f + (I - T_C) f.

Dove:

- T_C f è componente visibile.
- (I - T_C) f è componente invisibile.

Il kernel di T_C è:

Ker(T_C) =
{ f : E[f | Z] = 0 }.

7.4 Interpretazione decisionale

Consideriamo funzione di perdita:

ell(theta,a,X).

Definiamo rischio:

R(theta,a) = E[ell(theta,a,X)].

La funzione:

f_{theta,a}(X) = ell(theta,a,X)

appartiene a L2.

La decisione ottima dipende dalla famiglia:

{ f_{theta,a} }_a.

7.5 Collasso funzionale

DEFINIZIONE 7.1

La compressione C induce collasso funzionale in theta se esiste a tale che:

(I - T_C) f_{theta,a} ≠ 0.

Cioè la perdita contiene componente invisibile.

Interpretazione:

La loss dipende da componenti della distribuzione non osservabili in Z.

7.6 Teorema funzionale

TEOREMA 7.1

Se esiste theta tale che per l'azione ottima a*(theta):

(I - T_C) f_{theta,a*(theta)} ≠ 0,

allora C è decisionalmente insufficiente in theta.

Dimostrazione

Se la componente invisibile è non nulla, allora esistono variazioni nella distribuzione di X che:

- non modificano Z,
- modificano l'aspettativa della loss.

Quindi esistono theta1, theta2 con stessa legge di Z ma rischio differente.

Per il Capitolo 2, segue incompletezza minimax.

QED.

7.7 Relazione con sufficienza statistica classica

Un statistico T è sufficiente se:

P_theta(X | T) non dipende da theta.

In termini funzionali:

ogni funzione f dipendente da theta è funzione di T.

Qui il criterio è più debole e task-dependent: non chiediamo sufficienza globale, ma solo preservazione delle componenti rilevanti per la loss.

7.8 Norme e perdita di energia

Consideriamo norma L2:

||f||^2 = E[f^2].

La quantità:

|| (I - T_C) f ||^2

misura quanta “energia decisionale” è persa.

Questa è una misura quantitativa del collasso funzionale.

7.9 Collegamento con divergenze

Se f rappresenta log-likelihood ratio:

f = log(dP_theta1 / dP_theta2),

allora:

|| (I - T_C) f ||^2

è collegata alla riduzione della KL-divergence.

Quindi il formalismo funzionale generalizza quello informazionale.

7.10 Struttura unificata

Ora abbiamo tre descrizioni equivalenti del fenomeno:

1. Discreta (collasso di stati).
2. Informazionale (contrazione divergenze).
3. Funzionale (proiezione ortogonale).

Tutte convergono allo stesso risultato:

Se la compressione elimina componenti rilevanti per la loss, l’incompletezza è inevitabile.

CAPITOLO 8

VERSIONE GENERALE SU SPAZI PARAMETRICI CONTINUI E OPERATORI NON LINEARI

8.1 Setup astratto generale

Sia:

Theta ⊂ R^d aperto, connesso.

Per ogni theta ∈ Theta, sia P_theta una famiglia regolare di distribuzioni su X.

Sia:

C : X -> Z

una compressione misurabile (non necessariamente lineare).

Sia:

ell : Theta x A x X -> R

una funzione di perdita tale che:

0 <= ell <= 1,
ell è C^1 in theta,
ell è misurabile in X.

Rischio:

R(theta,a) = E_{X~P_theta}[ell(theta,a,X)].

8.2 Operatore decisionale generale

Definiamo:

Gamma(theta,a) =
E_{X~P_theta}[ell(theta,a,X) | Z].

Il sistema basato su Z osserva solo Gamma.

Il rischio compresso è:

R_C(theta,a) = E[Gamma(theta,a)].

8.3 Fibrazione indotta dalla compressione

Definiamo relazione di equivalenza:

theta1 ~ theta2 se
P_{theta1}(Z) = P_{theta2}(Z).

Le classi di equivalenza sono fibre nello spazio parametrico.

Chiamiamo:

F_z = { theta : legge di Z è fissata }.

Queste fibre possono essere varietà di dimensione positiva.

8.4 Condizione globale di sufficienza decisionale

C è globalmente decisionalmente sufficiente se:

per ogni fibra F_z, l'azione ottima è costante su F_z.

Formalmente:

theta1, theta2 ∈ F_z

=> a*(theta1) = a*(theta2).

Se esiste fibra su cui a* non è costante, allora esiste collasso globale.

8.5 Teorema generale globale

TEOREMA 8.1 (Incompletezza globale su fibre)

Se esiste fibra F_z tale che:

a*(theta) non è costante su F_z,

allora:

inf_pi sup_theta regret(theta;pi) > 0.

Dimostrazione

Per ogni theta1, theta2 nella stessa fibra:

C(P_theta1) = C(P_theta2).

Quindi ogni policy pi assegna stessa azione su entrambi.

Se le azioni ottime differiscono, almeno uno dei due produce regret positivo.

Compattezza locale e continuità garantiscono bound epsilon > 0.

QED.

8.6 Caso non lineare generale

Finora C era mappa su X.

Possiamo estendere a operatori non lineari su distribuzioni:

C* : P_theta -> Q_theta

dove Q_theta è distribuzione su Z.

L'intero formalismo resta valido sostituendo C(X) con Q_theta.

La struttura non dipende dalla linearità.

Dipende solo dall'identificazione di stati distinti.

8.7 Formulazione operatore su misure

Sia M lo spazio delle misure su X.

C induce operatore:

C# : M -> M_Z

(pushforward).

Il collasso decisionale equivale a:

C#(P_theta1) = C#(P_theta2)
ma
R(theta1, a*(theta2)) ≠ R(theta1, a*(theta1)).

Questa è formulazione puramente misura-teorica.

8.8 Versione con varietà parametrica

Supponiamo che:

Theta sia varietà liscia di dimensione d.

Se la mappa:

theta -> C#(P_theta)

ha rango locale r < d,

allora le fibre hanno dimensione >= d - r.

Se la funzione:

theta -> a*(theta)

varia lungo direzioni della fibra, allora esiste collasso inevitabile.

Questo unifica:

- Teorema del rango costante,
- Analisi differenziale,
- Struttura globale.

8.9 Struttura minimax globale

Possiamo ora formulare il principio generale:

Se la dimensione effettiva osservabile tramite C è inferiore alla dimensione del sottospazio rilevante per la decisione, allora il regret minimax è strettamente positivo.

In formula concettuale:

dim(immagine C) < dim(sottospazio decisionale)
=> incompletezza.

8.10 Struttura universale del fenomeno

Ora abbiamo quattro formulazioni equivalenti:

1. Collasso discreto di stati.
2. Contrazione di divergenza informazionale.
3. Perdita di componente ortogonale funzionale.
4. Degenerazione del differenziale parametrico.

Questo indica che il fenomeno non è artificiale.

È strutturale rispetto a qualsiasi riduzione di rappresentazione.

8.11 Sintesi

Abbiamo costruito una teoria completa che afferma:

Ogni compressione che identifica stati decisionalmente distinti induce un limite minimax positivo non eliminabile.

Questo limite può essere:

- espresso tramite separazione di superfici di rischio,
- espresso tramite TV,
- espresso tramite KL,

- espresso tramite norma L2,
- espresso tramite rango differenziale.

CAPITOLO 9

FUNZIONI DI PERDITA GENERALI E REGRET STRUTTURALE AVANZATO

9.1 Setup generale

Sia:

Theta ⊂ R^d compatto.

Spazio azioni A compatto metrico.

Loss:

ell : Theta x A x X -> R

con le seguenti proprietà:

1. ell è misurabile in X.
2. ell è continua in (theta, a).
3. ell è L-Lipschitz in theta:
   |ell(theta1,a,x) - ell(theta2,a,x)| <= L ||theta1 - theta2||.
4. ell è convessa in a (quando A è convesso).

Rischio:

R(theta,a) = E_{P_theta}[ell(theta,a,X)].

9.2 Struttura del regret per loss generali

Regret puntuale:

regret(theta;pi) =
R(theta, a_pi(theta)) - R(theta, a*(theta)).

Non assumiamo struttura discreta.

L'azione ottima è:

a*(theta) ∈ argmin_a R(theta,a).

9.3 Condizione di separazione forte

DEFINIZIONE 9.1

La superficie di rischio è fortemente separata in theta se esiste m(theta) > 0 tale che:

per ogni a ≠ a*(theta):

R(theta,a) >= R(theta,a*(theta)) + m(theta).

Se m(theta) è uniformemente > 0 su un compatto, abbiamo separazione forte globale.

9.4 Teorema di incompletezza con separazione forte

TEOREMA 9.1

Supponiamo:

1. C induce collasso decisionale.

2. La superficie di rischio è fortemente separata.

Allora esiste epsilon >= m_min > 0 tale che:

inf_pi sup_theta regret(theta;pi) >= m_min.

Dimostrazione

Se su una fibra F_z esistono theta1, theta2 con azioni ottime diverse, per ogni policy pi si sceglie un'unica azione a_bar.

Se a_bar ≠ a*(theta1),
regret(theta1;pi) >= m(theta1).

Se a_bar ≠ a*(theta2),
regret(theta2;pi) >= m(theta2).

Quindi:

sup_theta regret(theta;pi) >= min(m(theta1),m(theta2)).

Se m è uniformemente > 0, segue bound globale.

QED.

9.5 Versione con convexity e strong convexity

Supponiamo ora che:

R(theta,a) sia mu-strongly convex in a.

Allora:

R(theta,a) - R(theta,a*(theta))

= (mu/2) || a - a*(theta) ||^2.

Se su una fibra a*(theta1) ≠ a*(theta2), la distanza tra azioni è positiva.

Sia:

delta_a = || a*(theta1) - a*(theta2) ||.

Allora per ogni policy:

epsilon >= (mu/2) (delta_a^2)/4.

Questo fornisce bound quadratico esplicito.

9.6 Caso Lipschitz in theta

Se:

R(theta,a) è L-Lipschitz in theta, e su una fibra F_z
||theta1 - theta2|| >= d,

allora variazione del rischio è almeno:

= L d.

Questo collega incompletezza a distanza parametrica.

9.7 Regret strutturale come funzione della fibra

Definiamo:

diam_dec(F_z) =
sup_{theta1,theta2 in F_z}
|| a*(theta1) - a*(theta2) ||.

Se diam_dec(F_z) > 0, allora esiste incompletezza.

Se R è strongly convex in a,
epsilon >= (mu/8) (diam_dec(F_z))^2.

9.8 Struttura globale

Possiamo ora formulare principio generale:

Se esiste fibra F_z con diam_dec(F_z) > 0, e la superficie di rischio soddisfa condizione di separazione regolare, allora il regret minimax è strettamente positivo.

Questo vale per:

- classificazione,
- regressione,
- decisione continua,
- problemi convex.

9.9 Unificazione con capitoli precedenti

Caso 0-1:
m(theta) = differenza di probabilità di errore.

Caso KL:
m(theta) collegato a divergenza.

Caso differenziale:
m(theta) collegato a gradiente.

Caso funzionale:
m(theta) collegato a componente ortogonale.

Tutte le forme si ricompongono in questo schema generale.

9.10 Struttura universale del limite

Il limite decisionale è funzione di tre quantità:

1. Geometria delle fibre indotte da C.
2. Separazione della superficie di rischio.
3. Regolarità della loss.

Formula concettuale generale:

epsilon >= Phi(
struttura_fibra,
separazione_rischio,
regolarità_loss
)

9.11 Significato profondo

Non è la media a causare incompletezza.
È la perdita di dimensioni decisionali.

Se la compressione riduce lo spazio osservabile sotto la dimensione rilevante per la decisione, il limite è inevitabile.

CAPITOLO 10

REGRET MINIMAX SU CLASSI DI DISTRIBUZIONI

10.1 Dalla parametrizzazione alla classe astratta

Finora abbiamo considerato:

P_theta, theta ∈ Theta.

Ora consideriamo una classe astratta di distribuzioni:

P ⊂ M(X),

dove M(X) è lo spazio delle misure di probabilità su X.

Non assumiamo parametrizzazione finita.

Il problema decisionale diventa:

inf_pi sup_{P ∈ P} R(P, pi) - R*(P),

dove:

R(P,a) = E_P[ell(P,a,X)],
R*(P) = inf_a R(P,a).

10.2 Compressione come operatore su misure

La compressione induce pushforward:

C# : M(X) -> M(Z),

C#(P) = legge di Z quando X ~ P.

Definiamo equivalenza:

P ~ Q se C#(P) = C#(Q).

Le classi di equivalenza sono fibre nel dominio delle misure.

10.3 Definizione di collasso strutturale su classe

DEFINIZIONE 10.1

La compressione C induce collasso strutturale su P se esistono:

P1, P2 ∈ P tali che:

C#(P1) = C#(P2)
ma
a*(P1) ≠ a*(P2).

Qui a*(P) è l'azione ottima per distribuzione P.

10.4 Teorema generale su classi

TEOREMA 10.1 (Incompletezza su classi)

Se C induce collasso strutturale su P,
allora:

inf_pi sup_{P ∈ P} regret(P; pi) > 0.

Dimostrazione

Identica alla versione parametrica:

Se due distribuzioni sono indistinguibili dopo compressione, ogni policy pi assegna stessa azione.

Poiché azioni ottime differiscono, almeno una produce regret positivo.

Compattezza o chiusura della classe garantisce bound positivo.

QED.

10.5 Avversario minimax

Ora interpretiamo il sup come avversario.

L'avversario sceglie P ∈ P dopo aver visto la policy.

Se esiste fibra con azioni ottime differenti, l'avversario può sempre scegliere la distribuzione che massimizza il regret.

Quindi l'incompletezza è robusta rispetto all'ordine di gioco.

10.6 Versione robusta con classi di distribuzioni vicine

Supponiamo ora che:

P sia classe epsilon-perturbazione di una famiglia base.

Se la compressione elimina dimensioni rilevanti, l'avversario può sfruttare variazioni invisibili.

Questo collega incompletezza a robustezza distribution-shift.

10.7 Relazione con teoria di Blackwell

In teoria delle decisioni statistiche:

Esperimento E1 è più informativo di E2 se per ogni decision problem, il rischio minimo con E1 è <= quello con E2.

Qui:

Se C non è Blackwell-sufficiente, allora esiste decision problem per cui il rischio aumenta.

Il nostro teorema mostra qualcosa di più forte:

Se C collassa stati decisionalmente distinti, allora l'aumento è strettamente positivo in minimax.

10.8 Bound generale su classi

Sia:

diam_dec(F) =
sup_{P,Q in fibra}
|| a*(P) - a*(Q) ||.

Se la loss è strongly convex, allora:

epsilon >= (mu/8) (diam_dec(F))^2.

Questo è identico alla versione parametrica, ma ora in spazio delle misure.

10.9 Struttura unificata

Ora abbiamo:

- versione discreta,
- versione continua,
- versione differenziale,
- versione funzionale,
- versione informazionale,
- versione su classi astratte.

Tutte producono lo stesso fenomeno:

La compressione che identifica distribuzioni decisionalmente distinte induce limite minimax positivo.

10.10 Significato epistemologico emergente

Non è un fatto tecnico.

È un principio generale:

La riduzione di rappresentazione crea inevitabilmente zone decisionali non distinguibili.

Se in quelle zone la decisione ottima varia, esiste un limite non eliminabile.

CAPITOLO 11

PRINCIPIO DI DIMENSIONE DECISIONALE

11.1 Motivazione

In tutti i capitoli precedenti è emerso un fatto ricorrente:

La compressione elimina informazione.

Ma non tutta l'informazione è rilevante per la decisione.

La domanda fondamentale è:

Quanta dimensione decisionale è necessaria per preservare ottimalità?

11.2 Definizione di dimensione decisionale effettiva

Sia:

Theta ⊂ R^d.

Definiamo mappa delle azioni ottime:

a* : Theta -> A.

Definiamo:

D_dec = dim(Im(a*)).

Interpretazione:

D_dec misura la dimensione del sottospazio parametrico lungo cui la decisione varia.

Se a*(theta) è costante,
D_dec = 0.

Se varia lungo k direzioni indipendenti,
D_dec >= k.

11.3 Dimensione osservabile

Sia:

Phi : Theta -> spazio delle leggi di Z.

Definiamo:

D_obs = dim(Im(Phi)).

Se C riduce informazione,
D_obs <= d.

Se Phi è differenziabile,
D_obs = rank(DPhi).

11.4 Principio dimensionale

PRINCIPIO 11.1

Se

D_obs < D_dec,

allora esiste incompletezza decisionale strutturale.

11.5 Dimostrazione concettuale

Se la dimensione osservabile è inferiore alla dimensione lungo cui la decisione varia, allora esistono direzioni decisionali invisibili.

Queste direzioni generano fibre di equivalenza con azioni ottime differenti.

Per i capitoli precedenti, segue regret minimax positivo.

QED.

11.6 Versione differenziale formale

Supponiamo:

rank(DPhi(theta)) = r.

Supponiamo:

rank(D a*(theta)) = k.

Se r < k,
allora esistono v tali che:

DPhi(theta)[v] = 0
ma
D a*(theta)[v] ≠ 0.

Questo implica collasso differenziale.

11.7 Versione funzionale

Nel formalismo L2:

La compressione induce proiezione T_C.

Definiamo spazio decisionale:

S_dec = span{ f_{theta,a*(theta)} }.

Se:

dim(T_C(S_dec)) < dim(S_dec),

allora parte dello spazio decisionale è nel kernel.

Segue incompletezza.

11.8 Versione informazionale

Definiamo informazione decisionale:

I_dec = I(a*(Theta); X).

Se

I(a*(Theta); Z) < I_dec,

allora compressione non preserva variabilità decisionale.

Questo implica limite via Fano o TV.

11.9 Struttura universale

Possiamo ora formulare teorema sintetico:

TEOREMA 11.1 (Principio Universale)

Sia C una compressione.

Se C non preserva la dimensione effettiva del problema decisionale,
allora:

inf_pi sup_theta regret(theta;pi) > 0.

La preservazione richiede:

- fibre con azione costante,
  oppure
- dimensione osservabile >= dimensione decisionale.

11.10 Significato profondo

La teoria può essere riassunta in una singola frase strutturale:

La decisione richiede una certa dimensione effettiva.
La compressione che riduce tale dimensione genera incompletezza inevitabile.

Non è un fenomeno statistico.
Non è un fenomeno numerico.
È un fenomeno strutturale.

CAPITOLO 12

FORMALIZZAZIONE ASSIOMATICA COMPLETA

12.1 Struttura primitiva

Introduciamo gli oggetti primitivi:

- Theta: insieme non vuoto (spazio degli stati).
- A: insieme non vuoto (spazio delle azioni).
- P: classe di distribuzioni su X.
- ell: funzione di perdita.
- C: operatore di compressione.
- pi: policy decisionale su Z.

Non assumiamo inizialmente struttura topologica o differenziale.

12.2 Assiomi fondamentali

ASSIOMA A1 (Esistenza del rischio)

Per ogni P ∈ P e a ∈ A, esiste:

R(P,a) = E_P[ell(P,a,X)].

ASSIOMA A2 (Ottimalità locale)

Per ogni P ∈ P esiste almeno un:

a*(P) ∈ argmin_a R(P,a).

ASSIOMA A3 (Regret non negativo)

Per ogni P e a:

R(P,a) >= R(P,a*(P)).

ASSIOMA A4 (Compressione)

Esiste mappa:

C# : P -> P_Z

che associa a ogni distribuzione P la distribuzione osservabile dopo compressione.

ASSIOMA A5 (Policy dipende solo da C#)

Ogni policy pi è funzione solo di C#(P).

Formalmente:

Se C#(P1) = C#(P2),
allora pi(P1) = pi(P2).

12.3 Definizione assiomatica di collasso

DEFINIZIONE 12.1

C induce collasso strutturale se esistono P1, P2 ∈ P tali che:

1. C#(P1) = C#(P2),

2. $a^*(P1) \neq a^*(P2)$.

12.4 Lemma fondamentale

LEMMA 12.1

Se C induce collasso strutturale, allora per ogni policy pi esiste almeno uno tra P1, P2 tale che:

$R(P_i, pi) > R(P_i, a^*(P_i))$.

Dimostrazione

Per A5:

pi(P1) = pi(P2) = a_bar.

Se a_bar = a*(P1),
allora per P2 l'azione è subottima.

Se a_bar = a*(P2),
allora per P1 è subottima.

Per A3, regret positivo.

QED.

12.5 Teorema assiomatico principale

TEOREMA 12.1 (Incompletezza strutturale)

Se valgono A1–A5 e C induce collasso strutturale, allora:

$\inf_{pi} \sup_{P \in P} regret(P; pi) > 0$.

Dimostrazione

Dal Lemma 12.1, per ogni pi esiste P_i con regret positivo.

Se la classe P è finita, esiste minimo positivo.

Se P è compatta e R continua, il sup è raggiunto.

In ogni caso il regret minimax è strettamente positivo.

QED.

12.6 Carattere minimale del sistema

Osservazione cruciale:

Il teorema non richiede:

- linearità,
- differenziabilità,
- convexità,
- informazione mutua,
- KL,
- TV,
- topologia.

È pura conseguenza logica di:

- identificazione di stati,
- differenza di azioni ottime.

Questo mostra che il fenomeno è assiomaticamente inevitabile.

12.7 Struttura di completezza

Possiamo ora definire:

C è decisionalmente completo se e solo se non esiste collasso strutturale.

Equivalentemente: per ogni fibra di C#, l'azione ottima è costante.

12.8 Sistema compatto di teoremi

Abbiamo ora:

Teorema 12.1 — versione puramente logica.
Teorema 2.1 — versione topologica.
Teorema 4.1 — versione TV.
Teorema 5.1 — versione KL.
Teorema 6.1 — versione differenziale.
Teorema 7.1 — versione funzionale.
Teorema 11.1 — versione dimensionale.

Tutti derivano dalla stessa struttura assiomatica.

12.9 Conclusione del blocco assiomatico

Il limite decisionale non è un artefatto tecnico.

È una conseguenza formale del fatto che: una funzione che identifica elementi distinti non può preservare una mappa che li distingue.

In forma puramente matematica:

Se $C \circ a^*$ non è iniettiva, l'ottimalità non è preservabile globalmente.

CAPITOLO 13

INCOMPLETEZZA STRUTTURALE DEI SISTEMI DECISIONALI BASATI SU MEDIE

13.1 Motivazione

Molti sistemi decisionali reali si basano su rappresentazioni aggregate:

- media,
- vettore di momenti,
- embedding ridotto,
- score sintetico.

Formalmente, tali sistemi operano tramite una mappa:

T : X -> R^k,

con k << dimensione effettiva di X.

Chiamiamo questi:

Sistemi decisionali a rappresentazione compressa finita.

13.2 Definizione di sistema decisionale basato su medie

DEFINIZIONE 13.1

Un sistema è basato su medie se la decisione dipende solo da:

T(X) = E[phi(X)],

per un insieme finito di funzioni phi_1, ..., phi_k.

In altre parole, il sistema utilizza solo k momenti o statistiche riassuntive.

13.3 Struttura parametrica non lineare

Supponiamo che lo spazio degli stati sia:

Theta ⊂ R^d,

con d > k.

Se la decisione ottima varia lungo tutte le d dimensioni, ma il sistema osserva solo k momenti, allora esistono d - k direzioni invisibili.

Questo è un fatto dimensionale.

13.4 Teorema di incompletezza per sistemi basati su medie

TEOREMA 13.1

Sia:

T : X -> R^k

una rappresentazione finita.

Supponiamo che esistano theta1 != theta2 tali che:

E_{P_theta1}[phi_i(X)] = E_{P_theta2}[phi_i(X)]
per ogni i = 1,...,k,

ma

a*(theta1) ≠ a*(theta2).

Allora:

inf_pi sup_theta regret(theta;pi) > 0.

Dimostrazione

Le distribuzioni P_theta1 e P_theta2 producono stessa rappresentazione T.

Ogni policy basata su T assegna stessa azione.

Per il sistema assiomatico del Capitolo 12, segue incompletezza.

QED.

13.5 Interpretazione strutturale

Un sistema basato su medie è un sistema che opera su una proiezione finita dello spazio delle distribuzioni.

Se il problema decisionale richiede più dimensioni di quelle catturate dalla rappresentazione, esistono stati decisionalmente distinti ma indistinguibili dal sistema.

Questa è incompletezza decisionale.

13.6 Versione continua quasi-gödeliana

In analogia strutturale:

In un sistema formale, se il linguaggio ha potenza espressiva limitata, esistono verità non esprimibili.

Qui:

Se la rappresentazione ha dimensione limitata, esistono stati decisionali non distinguibili.

La struttura è:

Limitazione rappresentativa
⇒ perdita di distinguibilità
⇒ incompletezza rispetto alla decisione ottima.

Non è incompletezza logica formale, ma incompletezza rispetto alla classe dei compiti decisionali.

13.7 Forma astratta del principio

Sia:

R: P -> A

la mappa ottimale.

Sia:

T: P -> R^k

la rappresentazione.

Se esiste fibra F di T tale che:

R non è costante su F, allora T è incompleto rispetto a R.

Questo è puramente funzionale.

13.8 Conseguenza generale

Ogni sistema decisionale che opera su:

- numero finito di momenti,
- embedding ridotto,
- media globale,

è potenzialmente incompleto per classi sufficientemente ricche di problemi.

Non è questione di dati insufficienti.
È questione di struttura rappresentativa.

13.9 Collegamento ai capitoli precedenti

Questo risultato è una conseguenza diretta di:

- Principio dimensionale (Capitolo 11),
- Teorema assiomatico (Capitolo 12),
- Geometria differenziale (Capitolo 6).

La versione "basata su medie" è un caso particolare del principio generale.

13.10 Chiusura del capitolo

Possiamo ora formulare il risultato in forma sintetica:

Ogni sistema decisionale fondato su una rappresentazione compressa di dimensione inferiore alla dimensione effettiva del problema è strutturalmente incompleto rispetto alla decisione ottima.

Questo conclude la parte teorica fondazionale.

CAPITOLO 14

TEOREMA UNIVERSALE DI INCOMPLETEZZA CON BOUND STRUTTURALE

14.1 Obiettivo

Vogliamo una forma completamente generale del risultato che:

- non dipenda dal caso binario,
- non dipenda da convexità specifica,
- non dipenda da finitezza di Theta,
- non dipenda da loss 0-1,

ma che produca un bound inferiore esplicito in funzione di:

1. Geometria delle fibre,
2. Separazione della superficie di rischio,
3. Regolarità locale.

14.2 Setup generale

Sia:

Theta compatto metrico.
A compatto metrico.

R : Theta x A -> R continua.

C# : Theta -> Y continua.

Fibre:

F_y = { theta ∈ Theta : C#(theta) = y }.

Definiamo funzione ottima:

a*(theta) ∈ argmin_a R(theta,a).

14.3 Quantità strutturale fondamentale

Definiamo per ogni fibra F_y:

Delta(F_y) =
inf_{theta1,theta2 in F_y}
inf_{a}
max(
R(theta1,a) - R(theta1,a*(theta1)),
R(theta2,a) - R(theta2,a*(theta2))
).

Questa è la minima penalità inevitabile su quella fibra.

14.4 Teorema universale

TEOREMA 14.1

Supponiamo esista fibra F_y tale che:

1. a*(theta) non è costante su F_y,

2. R è continua,

3. Theta compatto.

Allora:

inf_pi sup_theta regret(theta;pi)

=

min_{F_y con collasso} Delta(F_y)

0.

14.5 Dimostrazione completa

1. Sia F_y fibra con collasso.
   Esistano theta1, theta2 ∈ F_y tali che a*(theta1) ≠ a*(theta2).

2. Ogni policy pi dipende solo da y, quindi assegna stessa azione a_bar su F_y.

3. Per definizione di Delta(F_y):

max(
R(theta1,a_bar) - R(theta1,a*(theta1)),
R(theta2,a_bar) - R(theta2,a*(theta2))
)

= Delta(F_y).

4. Per continuità e compattezza, Delta(F_y) > 0.

5. Il sup su theta è >= Delta(F_y).

6. L’inf su pi non può scendere sotto tale quantità.

QED.

14.6 Struttura del bound

Il bound dipende solo da:

- differenza tra azioni ottime sulla fibra,
- separazione della superficie di rischio.

Non dipende dalla specifica policy.

È universale rispetto a tutte le policy basate su C#.

14.7 Versione differenziale esplicita

Se Theta è varietà liscia e R C^2, supponiamo esista v tale che:

DC#(theta)[v] = 0

e

grad_theta R(theta,a*(theta)) · v ≠ 0.

Allora per piccolo t:

Delta(F_y) >=

| grad_theta R(theta,a*(theta)) · v | * t

O(t^2).

Quindi epsilon cresce linearmente con variazione invisibile.

14.8 Versione strongly convex

Se R(theta,a) è mu-strongly convex in a,

e su fibra:

|| a*(theta1) - a*(theta2) || >= delta_a,

allora:

Delta(F_y) >= (mu/8) delta_a^2.

14.9 Forma compatta finale

Possiamo scrivere:

epsilon >=
inf_{F_y con collasso}
Psi(F_y),

dove Psi dipende da:

- separazione delle superfici,
- regolarità,
- struttura geometrica.

Questo è il bound universale più generale coerente con la teoria.

14.10 Conseguenza forte

Il limite decisionale non è:

- un errore algoritmico,

- un problema di stima,
- una questione di campioni.

È funzione della struttura della rappresentazione.

La compressione induce un limite deterministico strutturale.

CAPITOLO 15

SISTEMI ADATTIVI, FEEDBACK E LIMITE STRUTTURALE DINAMICO

15.1 Motivazione

Finora abbiamo analizzato sistemi statici:

- un singolo stato theta,
- una singola decisione,
- una singola osservazione.

Ora consideriamo un sistema dinamico sequenziale:

Al tempo t = 1,2,...,T:

1. Si osserva X_t ~ P_theta.
2. Si osserva Z_t = C(X_t).
3. Si sceglie azione a_t.
4. Si osserva loss ell(theta, a_t, X_t).

Il sistema può adattarsi nel tempo.

Domanda: Può l'adattività eliminare il limite strutturale?

15.2 Setup dinamico

Sia:

pi_t : Z_1,...,Z_t -> A

una policy adattiva.

Regret cumulativo:

Regret_T(theta; pi) =
sum_{t=1}^T
[R(theta, a_t) - R(theta, a*(theta))].

Regret medio:

(1/T) Regret_T.

15.3 Definizione di collasso dinamico

C induce collasso dinamico se esistono theta1, theta2 tali che:

per ogni sequenza di osservazioni Z_1,...,Z_T, la distribuzione è identica sotto P_theta1 e P_theta2.

Formalmente:

C#(P_theta1) = C#(P_theta2).

Questo implica che nessuna strategia adattiva può distinguere theta1 da theta2.

15.4 Lemma di indistinguibilità sequenziale

LEMMA 15.1

Se C#(P_theta1) = C#(P_theta2), allora per ogni T e per ogni policy adattiva:

la distribuzione della sequenza Z_1,...,Z_T è identica sotto theta1 e theta2.

Dimostrazione

Poiché ogni Z_t dipende solo da C(X_t), e X_t sono iid sotto P_theta, la legge congiunta è prodotto delle leggi marginali.

Se le marginali coincidono, le leggi congiunte coincidono.

QED.

15.5 Teorema dinamico principale

TEOREMA 15.1 (Incompletezza dinamica)

Se esiste collasso strutturale tra theta1 e theta2, allora per ogni policy adattiva pi_T:

liminf_{T→∞}
sup_{theta ∈ {theta1,theta2}}
(1/T) Regret_T(theta;pi_T)

= epsilon > 0.

Dimostrazione

Per Lemma 15.1, la policy produce stessa sequenza di azioni sotto theta1 e theta2.

Sia sequenza a_1,...,a_T.

Per uno dei due stati, le azioni non coincidono con a*(theta).

Poiché separazione della superficie di rischio è positiva, esiste m > 0 tale che:

R(theta,a) - R(theta,a*(theta)) >= m.

Quindi per uno dei due theta:

Regret_T(theta;pi_T) >= mT.

Dividendo per T:

Regret medio >= m.

QED.

15.6 Significato

Nemmeno osservando infiniti campioni Z_t si può eliminare il limite se la rappresentazione non distingue gli stati.

L'adattività non crea informazione.

Se l'informazione non è presente in Z, nessuna strategia può ricostruirla.

15.7 Versione informazionale sequenziale

Per T osservazioni:

I(Theta; Z_1,...,Z_T)
<= T I(Theta; Z_1).

Se I(Theta; Z_1) = 0 tra theta1 e theta2, allora nessun numero di osservazioni produce separazione.

Quindi Fano dinamico produce stesso limite.

15.8 Caso con feedback completo

Anche se l'agente osserva loss passate, la loss è funzione di X_t e a_t.

Se Z_t non distingue gli stati, e la loss osservata non rivela informazione aggiuntiva, il collasso resta.

15.9 Struttura universale dinamica

Il principio dinamico è:

Se la rappresentazione elimina informazione strutturalmente rilevante, l'apprendimento non può recuperarla.

La compressione crea un vincolo permanente.

15.10 Sintesi teorica globale

Ora abbiamo dimostrato incompletezza in:

1. Versione statica.
2. Versione quantitativa.
3. Versione informazionale.
4. Versione differenziale.
5. Versione funzionale.
6. Versione su classi.

7. Versione dimensionale.

8. Versione assiomatica.

9. Versione dinamica adattiva.

Questo chiude il blocco teorico centrale.

CAPITOLO 16

ARCHITETTURA DI FORMALIZZAZIONE MACHINE-CHECKED

16.1 Obiettivo

Tradurre il sistema assiomatico e i teoremi principali in:

- definizioni Lean formalmente tipizzate
- teoremi verificabili
- struttura modulare
- assenza di ambiguità logica
- verificabilità automatica

Questo capitolo definisce la struttura di formalizzazione completa.

16.2 Struttura logica di base (Lean blueprint)

16.2.1 Tipi primitivi

In Lean:

variable {Theta : Type}

variable {Action : Type}

variable {X : Type}

Spazio delle distribuzioni:

variable (P : Theta → ProbabilityMeasure X)

Loss:

variable (ell : Theta → Action → X → ℝ)

Rischio:

def Risk (θ : Theta) (a : Action) : ℝ :=

∫ x, ell θ a x ∂(P θ)

16.2.2 Azione ottima

def OptimalAction (θ : Theta) : Action :=

argmin (fun a => Risk θ a)

Richiede:

- Action finito o compatto
- continuità di Risk

16.2.3 Compressione

variable (C : X → Z)

Pushforward:

def PushForward (θ : Theta) : ProbabilityMeasure Z :=

(P θ).map C

16.3 Definizione formale di collasso

def Collapsed (θ1 θ2 : Theta) : Prop :=

PushForward θ1 = PushForward θ2

∧ OptimalAction θ1 ≠ OptimalAction θ2

16.4 Teorema di incompletezza formalizzato

theorem structural_incompleteness

(h : ∃ θ1 θ2, Collapsed θ1 θ2) :

∃ ε > 0,

∀ π : Z → Action,

∃ θ,

Risk θ (π (C_sample θ)) - Risk θ (OptimalAction θ) ≥ ε

Nota:
C_sample θ formalizza campione distribuito secondo PushForward θ.

16.5 Livelli di formalizzazione

La formalizzazione completa richiede:

1. MeasureTheory
2. Integration
3. Topology

4. Compactness
5. Continuity
6. Argmin existence
7. Finite Action case iniziale
8. Estensione compatta

16.6 Blueprint strutturale repository

Struttura repository:

/DecisionIncompleteness

/Core

Assumptions.lean

Risk.lean

Compression.lean

Collapse.lean

/Minimax

Regret.lean

MinimaxTheorem.lean

/Information

TVBound.lean

KLBound.lean

/Differential

RankCollapse.lean

/Dynamic

SequentialLimit.lean

16.7 Gerarchia dei teoremi formalizzati

Livello 1 — Assiomatico puro

- structural_incompleteness

Livello 2 — Versione compatta continua

- minimax_positive_bound

Livello 3 — Bound TV

- tv_regret_lower_bound

Livello 4 — Bound KL

- kl_regret_lower_bound

Livello 5 — Versione differenziale

- rank_deficiency_implies_collapse

Livello 6 — Versione dinamica

- sequential_regret_lower_bound

16.8 Obiettivo machine-check finale

Un sistema completamente formalizzato deve garantire:

- Zero ambiguità semantica
- Dipendenze matematiche esplicite
- Tracciabilità delle ipotesi
- Verifica indipendente

16.9 Livello Coq alternativo

In Coq:

- Struttura typeclass per rischio
- Definizione di Collapsed come relation
- Teorema principale come lemma costruttivo
- Uso di Coquelicot per analisi

16.10 Certificazione esterna

Workflow:

1. Lean build
2. CI automatico
3. Tag versione
4. SWHID archivio Software Heritage

5. Manifest SHA256

Questo rende la teoria:

- auditabile
- replicabile
- non ambigua

16.11 Implicazione epistemologica formale

Una teoria formalizzata:

- non dipende dall'autore
- non dipende dall'interpretazione
- è verificabile simbolicamente

Il contenuto diventa indipendente dalla retorica.

CAPITOLO 17

PROGRAMMA EMPIRICO TOTALE E REPLICABILITÀ 100%

17.1 Obiettivo

Dimostrare empiricamente che:

Se una compressione collassa stati decisionalmente distinti, allora esiste regret minimo positivo misurabile.

Il programma deve:

- essere preregistrabile
- essere eseguibile
- produrre numeri
- essere ripetibile
- essere auditabile

17.2 Struttura generale degli esperimenti

Tre livelli di verifica:

1. Esperimento sintetico controllato (parametrico)
2. Esperimento distribuzionale non lineare

3. Esperimento dinamico sequenziale

17.3 ESPERIMENTO 1 — Collasso di media

Setup

Theta = {theta1, theta2}

Distribuzioni:

P_theta1: N(+mu, sigma^2)
P_theta2: N(-mu, sigma^2)

Compressione:

Z = E[X] (media stimata)

Decisione:

a = sign(media)

Costruzione modificata: Costruiamo distribuzioni con stessa media ma varianza differente tali che decisione ottima differisce.

Codice Python replicabile

```
import numpy as np


def simulate(theta, n=1000):
    if theta == 1:
        return np.random.normal(0, 1, n)
    else:
        return np.random.normal(0, 4, n)


def compress(x):
    return np.mean(x)


def optimal_action(theta):
    return 0 if theta == 1 else 1


def regret(theta, action):
    return 1 if action != optimal_action(theta) else 0


T = 10000
regrets = []
```

```
for _ in range(T):
    for theta in [1,2]:
        x = simulate(theta)
        z = compress(x)
        action = 0 if z < 0 else 1
        regrets.append(regret(theta, action))

print("Mean regret:", np.mean(regrets))
```

Risultato atteso: regret > 0.

17.4 ESPERIMENTO 2 — Bound TV

Stimiamo empiricamente:

```
TV(P_theta1, P_theta2)

def estimate_tv(samples1, samples2, bins=50):
    hist1,_ = np.histogram(samples1, bins=bins, density=True)
    hist2,_ = np.histogram(samples2, bins=bins, density=True)
    return 0.5*np.sum(np.abs(hist1-hist2))
```

Confrontiamo:

TV_X vs TV_Z

Verifichiamo:

Regret ≈ (1/2)(TV_X - TV_Z)

17.5 ESPERIMENTO 3 — Differenziale

Costruiamo modello continuo:

R(theta,a) = (a - theta)^2

Compressione:

Z = round(theta)

Simuliamo:

```
def risk(theta,a):
    return (a-theta)**2
```

Se round(theta1) = round(theta2) ma theta1 ≠ theta2, azione ottima a=theta non distinguibile.

Bound quadratico verificabile.

17.6 ESPERIMENTO 4 — Dinamico sequenziale

Simuliamo policy adattiva:

```
for t in range(T):
    observe Z_t
    update belief
    choose action
```

Se Z non distingue, regret medio converge a costante positiva.

17.7 JSON configurazione esperimenti

```
{
  "theta_space": "continuous",
  "compression": "mean",
  "loss": "quadratic",
  "samples": 10000,
  "repetitions": 50,
  "seed": 42
}
```

17.8 Manifest replicabilità

File:

- experiment1.py
- experiment2.py
- config.json
- results.csv
- manifest.sha256

SHA256 garantisce integrità.

17.9 Criteri di falsificazione

La teoria è falsificata se esiste:

- collasso strutturale,
- ma regret empirico → 0 con T → infinito.

Oppure:

- compressione elimina divergenza,
- ma non aumenta errore minimo.

Finora nessun controesempio noto coerente con assiomi.

17.10 Infrastruttura hardware

Requisiti minimi:

- Python 3.10+
- NumPy
- SciPy
- 8GB RAM
- CPU standard

Non richiede GPU.

17.11 Protocollo preregistrato

1. Definire distribuzioni
2. Definire compressione
3. Definire loss
4. Calcolare azione ottima teorica
5. Stimare regret
6. Confrontare con bound teorico

Tutti parametri fissati prima dell'esecuzione.

17.12 Audit indipendente

Un revisore deve poter:

- scaricare repository
- eseguire script
- verificare output
- controllare hash

Nessun parametro nascosto.

17.13 Conclusione empirica

Il fenomeno non è solo teorico.

È:

- osservabile,
- misurabile,
- ripetibile,
- robusto.

CAPITOLO 18

COLLOSSO COME PERDITA DI COMPONENTE ORTOGONALE

18.1 Riformulazione funzionale globale

Consideriamo spazio di Hilbert:

H_theta = L2(P_theta)

con prodotto interno:

⟨f,g⟩ = E_{P_theta}[f(X) g(X)]

Sia:

T_C : H_theta -> H_theta

l'operatore di proiezione:

T_C f = E[f | Z]

È ortogonale su sottospazio:

H_Z = { f misurabili rispetto a sigma(Z) }

18.2 Spazio decisionale effettivo

Definiamo per ogni theta:

S_theta = span{ f_{theta,a} : a ∈ A }

dove:

f_{theta,a}(X) = ell(theta,a,X)

Lo spazio decisionale effettivo globale è:

S = closure(∪_theta S_theta)

18.3 Definizione formale di collasso funzionale globale

DEFINIZIONE 18.1

C induce collasso funzionale se:

esiste theta tale che

(I - T_C) f_{theta,a*(theta)} ≠ 0.

Cioè la componente invisibile non è nulla.

18.4 Lemma fondamentale L2

LEMMA 18.1

Se

|| (I - T_C) f ||_2 > 0,

allora esistono P1, P2 nella stessa fibra tali che:

E_{P1}[f] ≠ E_{P2}[f].

Dimostrazione

Se la componente ortogonale è non nulla, allora f non è funzione di Z.

Quindi esistono punti X1, X2 tali che:

C(X1) = C(X2)
ma
f(X1) ≠ f(X2).

Costruendo misure concentrate su tali punti, si ottiene differenza di aspettativa con stessa legge di Z.

QED.

18.5 Teorema funzionale generale

TEOREMA 18.1

Se esiste theta tale che:

|| (I - T_C) f_{theta,a*(theta)} ||_2 > 0,

allora C è decisionalmente incompleto.

Dimostrazione

Dal Lemma 18.1 esistono P1, P2 con stessa legge Z ma rischio differente.

Per sistema assiomatico (Capitolo 12), segue regret minimax positivo.

QED.

18.6 Quantificazione tramite norma L2

Possiamo definire:

gamma_theta =
|| (I - T_C) f_{theta,a*(theta)} ||_2.

Allora:

epsilon >= c * gamma_theta^2

per costante c dipendente da regolarità della loss.

Questo collega norma funzionale al bound quantitativo.

18.7 Caso lineare

Se:

C è proiezione lineare su sottospazio V,

allora:

T_C è proiezione ortogonale su V.

Se spazio decisionale S non è contenuto in V, allora componente ortogonale non nulla.

Segue incompletezza.

18.8 Caso non lineare

Se C è non lineare,
T_C resta proiezione condizionale.

La decomposizione ortogonale rimane valida in L2.

Il risultato non dipende da linearità di C.

18.9 Unificazione definitiva delle formulazioni

Ora possiamo affermare equivalenza tra:

- Collasso discreto
- Perdita di divergenza
- Rank deficiency
- Perdita di componente ortogonale

Sono quattro manifestazioni dello stesso fenomeno.

18.10 Conclusione Parte V

La Parte V dimostra che:

Il limite strutturale è una conseguenza della perdita di componente funzionale rilevante nello spazio L2.

Questo chiude definitivamente la sezione operatoriale.

PARTE VI

ESTENSIONI GENERALI

CAPITOLO 19

PARAMETRI MULTI-DIMENSIONALI E STRUTTURA VETTORIALE

19.1 Motivazione

Finora abbiamo trattato:

Theta ⊂ R^d

ma spesso con focus implicito su singola coordinata rilevante.

Ora formalizziamo esplicitamente il caso:

theta = (theta_1, ..., theta_d) ∈ R^d

d > 1 reale e potenzialmente elevato.

Il punto centrale:

La decisione può dipendere solo da alcune componenti, mentre la compressione può ignorarne altre.

19.2 Decomposizione parametrica

Supponiamo:

Theta ⊂ R^d

con decomposizione ortogonale:

$R^d = V_{dec} \oplus V_{null}$

dove:

V_{dec} = sottospazio decisionale effettivo
V_{null} = sottospazio irrilevante per la decisione

Definizione:

Per ogni $v \in V_{null}$:

$a^*(\theta + v) = a^*(\theta)$.

19.3 Definizione di dimensione decisionale effettiva

DEFINIZIONE 19.1

$Dim_{dec} = \dim(V_{dec})$.

Questa è la dimensione minima del sottospazio parametrico che influenza l'azione ottima.

19.4 Compressione parametrica

Sia compressione parametrica:

$\Phi : \Theta \to Y \subset R^k$

con $k < d$.

$Rank(\Phi) = r <= k$.

Se:

r < Dim_dec

allora esistono direzioni decisionali invisibili.

19.5 Teorema multi-dimensionale

TEOREMA 19.1

Se rank(DPhi(theta)) < Dim_dec,
allora C è decisionalmente incompleto in un intorno di theta.

Dimostrazione

Esistono v ≠ 0 tali che:

DPhi(theta)[v] = 0

ma

D a*(theta)[v] ≠ 0.

Per Capitolo 6, segue collasso differenziale locale.

QED.

19.6 Caso lineare esplicito

Supponiamo:

theta ∈ R^d

Phi(theta) = A theta

con A matrice k x d.

Se rank(A) = k < Dim_dec,

allora esistono v nel kernel di A con:

a*(theta + v) ≠ a*(theta).

In tal caso incompletezza è inevitabile.

19.7 Bound quantitativo multi-dimensionale

Se R è mu-strongly convex in a, e lungo direzione invisibile v vale:

|| D a*(theta)[v] || >= alpha,

allora per piccolo t:

|| a*(theta + t v) - a*(theta) || >= alpha t.

Quindi:

epsilon >= (mu/8)(alpha^2 t^2).

19.8 Caso non lineare multi-parametrico

Se Phi è non lineare ma C^1, vale teorema del rango costante.

Se dimensione immagine < dimensione decisionale, allora fibre di dimensione >= Dim_dec - r > 0.

Se su tali fibre a* varia, segue incompletezza.

19.9 Interpretazione strutturale

La decisione vive su una varietà di dimensione Dim_dec.

La compressione vive su varietà di dimensione r.

Se r < Dim_dec, la rappresentazione non può parametrizzare completamente la decisione ottima.

Questo è un fatto puramente dimensionale.

19.10 Collegamento con capitoli precedenti

Questa estensione:

- generalizza Capitolo 6 (differenziale),
- generalizza Capitolo 11 (principio dimensionale),
- rafforza Capitolo 14 (bound universale).

19.11 Conclusione Capitolo 19

La teoria è stabile sotto aumento di dimensione.

Non è un artefatto 1D.

È un principio dimensionale generale.

CAPITOLO 20

PERDITA NON LIMITATA

20.1 Motivazione

Molti problemi reali non hanno loss limitata:

- Regressione quadratica: ell = (a − Y)^2
- Loss esponenziale
- Log-loss
- Funzioni polinomiali

In questi casi:

ell può essere non limitata superiormente. Dobbiamo dimostrare che il teorema di incompletezza resta valido.

20.2 Nuove ipotesi di regolarità

Assumiamo:

(A1) R(theta,a) finito per ogni theta,a
(A2) R continua in theta
(A3) Per ogni theta esiste a*(theta)
(A4) Separazione locale: esiste intorno U tale che
R(theta,a) − R(theta,a*(theta)) > 0 per a ≠ a*(theta)

Non assumiamo boundedness.

20.3 Regret con loss non limitata

Regret puntuale:

regret(theta;pi) =
R(theta, a_pi(theta)) − R(theta,a*(theta)).

Poiché R finito, regret è ben definito.

20.4 Teorema generale senza boundedness

TEOREMA 20.1

Supponiamo esista collasso strutturale:

C#(theta1) = C#(theta2)
a*(theta1) ≠ a*(theta2)

e R soddisfi separazione locale.

Allora:

inf_pi sup_theta regret(theta;pi) > 0.

Dimostrazione

Identica alla versione bounded:

Per ogni pi:

a_pi(theta1) = a_pi(theta2) = a_bar.

Se a_bar ≠ a*(theta1),

allora

regret(theta1;pi) = R(theta1,a_bar) − R(theta1,a*(theta1)) > 0.

Separazione locale garantisce minimo positivo.

Compattezza locale implica epsilon > 0.

QED.

20.5 Caso quadratico esplicito

Sia:

ell(theta,a,X) = (a − m(theta))^2

dove m(theta) = E_theta[X].

Supponiamo che compressione non distingua theta1 e theta2, ma m(theta1) ≠ m(theta2).

Allora:

a*(theta) = m(theta).

Per ogni policy basata su compressione, a_bar unico per entrambi.

Il regret minimo è:

min(
(m(theta1) − a_bar)^2,
(m(theta2) − a_bar)^2
)

Il minimo massimo si ottiene a_bar nel punto medio:

a_bar = (m(theta1)+m(theta2))/2.

Regret minimo = (1/4)(m(theta1) − m(theta2))^2.

Questo è bound quadratico esplicito non limitato.

20.6 Caso sub-quadratico generale

Se R è mu-strongly convex in a, allora per ogni a:

R(theta,a) − R(theta,a*(theta))
≥ (mu/2) ||a − a*(theta)||^2.

Questa proprietà non richiede boundedness.

Segue bound positivo identico al Capitolo 9.

20.7 Problema della divergenza infinita

Se ell non limitata, divergenze informazionali possono essere infinite.

Ma il teorema non richiede TV o KL.

È puramente strutturale.

Quindi la validità non dipende da integrabilità forte oltre R finito.

20.8 Stabilità sotto scaling

Se moltiplichiamo la loss per costante $c > 0$:

$\ell' = c\,\ell$,

allora:

$\text{regret}' = c\,\text{regret}$.

L'incompletezza scala linearmente.

20.9 Conclusione Capitolo 20

La boundedness non è essenziale.

Il limite strutturale vale per:

- regressione
- loss quadratiche
- log-loss
- funzioni polinomiali
- qualsiasi loss continua con minimo ben definito

CAPITOLO 21

DECISIONI CONTINUE

21.1 Setup generale

Sia:

Theta ⊂ R^d compatto

A ⊂ R^m convesso compatto

Loss:

ell : Theta × A × X → R

Rischio:

R(theta,a) = E_theta[ell(theta,a,X)]

Assumiamo:

1. R continua in (theta,a)
2. R C^1 in a
3. R mu-strongly convex in a

21.2 Azione ottima continua

Per ogni theta esiste unico minimizzatore:

a*(theta) ∈ A

caratterizzato da:

grad_a R(theta,a*(theta)) = 0

e matrice Hessiana:

H_a R(theta,a) ≥ mu I_m.

21.3 Collasso in regime continuo

Supponiamo esistano theta1, theta2 tali che:

C#(theta1) = C#(theta2)

ma

a*(theta1) ≠ a*(theta2).

Poiché policy dipende solo da C#, deve scegliere unico a_bar per entrambi.

21.4 Bound quadratico esplicito

Per mu-strong convexity:

R(theta,a) − R(theta,a*(theta))
≥ (mu/2) ||a − a*(theta)||^2.

Sia:

delta_a = || a*(theta1) − a*(theta2) ||.

Il minimax su scelta a_bar è ottenuto nel punto medio:

a_bar = (a*(theta1)+a*(theta2))/2.

Allora:

epsilon ≥ (mu/8) delta_a^2.

Questo è bound esplicito continuo.

21.5 Versione differenziale locale

Sia Theta continua e a*(theta) differenziabile.

Se esiste direzione v tale che:

DC#(theta)[v] = 0
D a*(theta)[v] ≠ 0,

allora per piccolo t:

|| a*(theta + t v) − a*(theta) || ≈ || D a*(theta)[v] || t.

Quindi:

epsilon ≥ (mu/8)(||D a*(theta)[v]||^2 t^2).

21.6 Caso non strongly convex

Se R è solo convessa ma non strongly convex, la separazione può essere lineare anziché quadratica.

In tal caso:

epsilon ≥ c || a*(theta1) − a*(theta2) ||, per qualche c > 0 derivabile dalla struttura della loss.

21.7 Regret integrale su dominio continuo

Se Theta è continuo e collasso avviene su regione U di misura positiva, allora:

sup_theta regret(theta;pi)
≥ inf_{theta∈U} separazione(theta).

Se separazione è continua e U compatto, esiste minimo positivo.

21.8 Caso regressione lineare

Esempio esplicito:

ell(theta,a,X) = (a − theta)^2

Compressione:

Phi(theta) = B theta

Se B ha rango r < d,
esistono theta1 ≠ theta2 con B theta1 = B theta2.

Bound:

epsilon ≥ (1/8)|| theta1 − theta2 ||^2.

21.9 Estensione a controllo ottimo

Consideriamo problema di controllo:

theta = parametro dinamico
a = controllo continuo

Se compressione elimina parametro rilevante per costo ottimo, allora costo minimo aumentato strutturalmente.

Teoria resta invariata.

21.10 Conclusione Capitolo 21

Il teorema di incompletezza vale integralmente per:

- azioni continue
- regressione
- controllo
- policy parametrizzate

La struttura non dipende dalla discrezione.

CAPITOLO 22

VERSIONE BAYESIANA

22.1 Setup bayesiano

Sia:

- Theta spazio parametrico
- π prior su Theta
- C : X → Z compressione
- a : Z → A policy

Rischio Bayes:

R_Bayes(a) =
∫_Theta R(theta, a(C#(theta))) dπ(theta)

Rischio Bayes ottimo completo (senza compressione):

R_Bayes^* =
∫_Theta R(theta, a*(theta)) dπ(theta)

Regret Bayes:

Regret_Bayes(a) =
R_Bayes(a) − R_Bayes^*.

22.2 Definizione di collasso bayesiano

Definiamo:

C induce collasso su prior π se esiste insieme S ⊂ Theta di misura positiva tale che:

1. C# costante su S
2. a*(theta) non costante su S

22.3 Teorema bayesiano principale

TEOREMA 22.1

Se C induce collasso su S con π(S) > 0 e R soddisfa separazione locale, allora:

inf_a Regret_Bayes(a) > 0.

Dimostrazione

Poiché C# costante su S, policy assegna unica azione a_bar su S.

Ma a*(theta) varia su S.

Quindi:

$$\int_S [R(\theta, \bar{a}) - R(\theta, a^*(\theta))] \, d\pi(\theta)$$
$$\geq$$
$$\pi(S) * \inf\{\theta \in S\} \text{ separazione}(\theta).$$

Se separazione continua e S compatto, esiste minimo positivo.

QED.

22.4 Bound esplicito strongly convex

Se R mu-strongly convex in a, e su S:

|| a*(theta1) − a*(theta2) || ≥ delta,

allora:

Regret_Bayes ≥
(μ/8) delta^2 π(S).

Bound lineare in massa prior.

22.5 Caso prior continuo

Se Theta continuo e collasso avviene lungo varietà di dimensione ≥ 1, e prior ha densità positiva su tale varietà, allora massa positiva inevitabile.

Il limite non dipende da prior "maligno".

22.6 Caso prior degeneri

Se prior concentra massa su punto singolo, ovviamente non esiste problema decisionale.

La teoria non pretende incompletezza in assenza di incertezza.

Questo è coerente.

22.7 Versione informazionale bayesiana

Mutual information:

$I(\Theta;Z) \leq I(\Theta;X)$.

Se compressione riduce informazione sotto soglia critica, posterior non distingue stati decisionalmente distinti.

Il regret Bayes minimo è limitato inferiormente da funzione di $I(\Theta;Z)$.

22.8 Bound tipo Fano bayesiano

Per testing binario indotto dal task:

$$\text{Regret}_{\text{Bayes}} \geq c \cdot P_{\text{error}}(Z),$$

dove P_{error} limitato inferiormente da Fano:

$$P_{\text{error}} \geq 1 - \frac{I(\Theta;Z)+\log 2}{\log|\Theta|}.$$

22.9 Interpretazione strutturale

Il limite bayesiano non è worst-case artificiale.

È conseguenza di:

- perdita strutturale di informazione decisionale
- impossibilità di distinguere stati con posterior identico

22.10 Conclusione Capitolo 22

Il teorema vale in:

- regime frequentista
- regime minimax
- regime bayesiano

Il limite è rappresentazionale, non metodologico.

CAPITOLO 23

VERSIONE ROBUSTA

23.1 Setup robusto

Sia:

Theta spazio parametrico.

Per ogni theta, non una singola distribuzione P_theta, ma un insieme:

P_theta ⊂ M(X)

(insieme di misure plausibili).

Rischio robusto:

R_rob(theta,a) =

sup_{P ∈ P_theta}

E_P[ell(theta,a,X)].

Azione ottima robusta:

a*_rob(theta) ∈ argmin_a R_rob(theta,a).

23.2 Compressione in regime robusto

Compressione:

C : X → Z

Induce pushforward:

C# P per ogni P ∈ P_theta.

Definiamo insieme compresso:

P_theta^Z =
{ C# P : P ∈ P_theta }.

23.3 Collasso robusto

Definizione:

C induce collasso robusto tra theta1 e theta2 se:

P_theta1^Z = P_theta2^Z

ma

a_*rob(theta1)* ≠ *a*_rob(theta2).

23.4 Teorema robusto principale

TEOREMA 23.1

Se esiste collasso robusto tra theta1 e theta2, e separazione robusta:

inf_a
max(
R_rob(theta1,a) − R_rob(theta1,a_*rob(theta1)),*
R_rob(theta2,a) − R_*rob(theta2,a*_rob(theta2))
)
= ε > 0,

allora:

inf_pi sup_theta R_rob(theta,pi) − R_rob(theta,a*_rob(theta))
≥ ε.

Dimostrazione

Poiché compressione non distingue insiemi robusti, policy deve assegnare stessa azione a_bar a theta1 e theta2.

Per definizione di separazione robusta, uno dei due ha regret ≥ ε.

QED.

23.5 Caso ambiguità Wasserstein

Supponiamo:

P_theta =
{ Q : W(Q,P_theta^0) ≤ rho }

dove W è distanza Wasserstein.

Se compressione elimina direzione rilevante, insiemi robusti compressi coincidono.

La perdita di informazione permane.

23.6 Caso avversariale

Consideriamo setting:

- Natura sceglie theta

- Avversario sceglie P ∈ P_theta
- Decision maker osserva Z

La compressione agisce prima dell'avversario.

Se collasso, nessuna strategia può distinguere stati robustamente distinti.

23.7 Regret robusto quadratico

Se R_rob è mu-strongly convex in a, e:

|| a_*rob(theta1)* − *a*_rob(theta2) || ≥ delta,

allora:

epsilon ≥ (mu/8) delta^2.

Stessa struttura del caso non robusto.

23.8 Interpretazione

Il limite non dipende da:

- stima puntuale
- modello corretto
- prior
- distribuzione singola

È proprietà della rappresentazione sotto incertezza.

23.9 Conclusione Parte VI

Abbiamo esteso la teoria a:

- parametri multi-dimensionali
- loss non limitate
- decisioni continue
- regime bayesiano
- regime robusto

Il fenomeno è strutturalmente stabile.

PARTE VII

REGRET STRUTTURALE AVANZATO

CAPITOLO 24

REGRET WORST-GROUP

24.1 Motivazione

Finora abbiamo analizzato:

sup_theta regret(theta;pi)

Ma in molti contesti (fairness, medicina, sistemi complessi) interessa:

Regret su sottogruppi.

Definiamo partizione:

Theta = ∪_g Theta_g

Regret worst-group:

Regret_wg(pi) =
max_g sup_{theta ∈ Theta_g}
regret(theta;pi)

24.2 Collasso su sottogruppi

Definizione:

C induce collasso su gruppo g se esistono theta1,theta2 ∈ Theta_g tali che:

C#(theta1) = C#(theta2)

ma

a*(theta1) ≠ a*(theta2).

24.3 Teorema worst-group

TEOREMA 24.1

Se esiste gruppo g con collasso strutturale, allora:

inf_pi Regret_wg(pi) > 0.

Dimostrazione

Limitandosi al gruppo g, il problema si riduce al teorema principale.

Poiché Regret_wg prende massimo sui gruppi, il bound positivo permane.

QED.

24.4 Bound esplicito worst-group

Se separazione su gruppo g è epsilon_g, allora:

inf_pi Regret_wg(pi) ≥ epsilon_g.

Se più gruppi hanno collasso, il bound è:

min_g epsilon_g.

24.5 Interpretazione

Il limite strutturale può colpire solo certi sottogruppi, ma questo è sufficiente per garantire incompletezza globale.

Questo collega la teoria a:

- fairness
- under-representation
- errori sistematici

CAPITOLO 25

MINIMAX SU CLASSI DI DISTRIBUZIONI

25.1 Setup

Sia classe:

F = { P_theta : theta ∈ Theta }

Definiamo:

R_F(a) =

sup_{P ∈ F}

E_P[ell(theta(P),a,X)].

Compressione C induce mapping:

F_Z = { C# P : P ∈ F }.

25.2 Collasso di classe

Definizione:

C induce collasso su F se esistono P1,P2 ∈ F tali che:

C# P1 = C# P2

ma

a*(P1) ≠ a*(P2).

25.3 Teorema classe

TEOREMA 25.1

Se esiste collasso su F, allora:

inf_pi sup_{P∈F}
regret(P;pi)

0.

Dimostrazione identica al caso parametrico, sostituendo Theta con F.

25.4 Caso distribuzioni non parametriche

Se F è insieme non parametrico (es. tutte distribuzioni con certe proprietà), la compressione può collassare sottoclasse decisionale rilevante.

Il risultato resta invariato.

CAPITOLO 26

BOUND UNIVERSALI

26.1 Obiettivo

Costruire bound indipendenti da dettagli specifici, dipendenti solo da grandezze strutturali.

26.2 Bound dimensionale universale

Sia:

Dim_dec dimensione decisionale effettiva
Dim_Z dimensione rappresentazione

Se Dim_Z < Dim_dec,
allora:

esiste epsilon ≥ c * delta^2, dove delta è separazione minima tra azioni ottime lungo fibre invisibili.

26.3 Bound informazionale universale

Se:

TV_X − TV_Z ≥ Delta,

allora:

epsilon ≥ (1/2) Delta.

26.4 Bound funzionale universale

Se:

gamma = || (I − T_C) f_{theta,a*} ||_2,

allora:

epsilon ≥ c gamma^2.

26.5 Bound differenziale universale

Se esiste v con:

DC[v] = 0
e
|| D a*(theta)[v] || ≥ alpha,

allora:

epsilon ≥ (mu/8) alpha^2 t^2.

26.6 Teorema unificato

TEOREMA 26.1 (Struttura Universale)

Sotto ipotesi di regolarità, le seguenti sono equivalenti:

1. Collasso discreto

2. Rank deficiency differenziale
3. Perdita di componente ortogonale L2
4. Perdita di divergenza informazionale
5. Bound positivo su regret minimax

Sono manifestazioni dello stesso fenomeno strutturale.

26.7 Conclusione Parte VII

Abbiamo mostrato che il limite:

- non è locale
- non è artificiale
- non è legato a una formulazione particolare

È strutturalmente universale.

PARTE VIII

PROGRAMMA EMPIRICO MONUMENTALE

CAPITOLO 27

COSTRUZIONE SIMULAZIONE STRUTTURALE

27.1 Obiettivo

Costruire una simulazione generale che permetta di:

1. Definire uno spazio parametrico Theta
2. Definire una famiglia di distribuzioni P_theta
3. Definire una compressione C
4. Calcolare l'azione ottima completa a*(theta)
5. Calcolare l'azione ottima compressa a_C(theta)
6. Misurare il regret strutturale

Il sistema deve essere:

- parametrico
- estendibile
- ripetibile
- auditabile

27.2 Struttura generale simulazione

Parametri principali:

- d_theta : dimensione parametro
- m_action : dimensione azione
- n_samples : numero campioni
- compression_type : {linear, nonlinear, projection, random}
- loss_type : {quadratic, logistic, custom}

27.3 Generatore distribuzioni parametriche

Caso base:

Theta ⊂ R^d

Distribuzione:

X | theta ~ N(theta, Sigma)

Implementazione:

```
import numpy as np


def generate_samples(theta, Sigma, n):
    return np.random.multivariate_normal(theta, Sigma, n)
```

27.4 Definizione loss generica

Quadratica:

```
def loss(theta, a, x):
    return np.sum((a - theta)**2)
```

Logistica:

```
def loss_logistic(theta, a, x):
    y = (x.dot(theta) > 0).astype(int)
    return - (y*np.log(a) + (1-y)*np.log(1-a))
```

27.5 Calcolo rischio empirico

```
def empirical_risk(theta, a, Sigma, n=10000):
    x = generate_samples(theta, Sigma, n)
    losses = [loss(theta, a, xi) for xi in x]
    return np.mean(losses)
```

27.6 Calcolo azione ottima completa

Nel caso quadratico:

a*(theta) = theta

In generale:

```
def optimal_action(theta):
    return theta
```

27.7 Compressione lineare

C(x) = B x

```
def compress_linear(x, B):
    return x.dot(B.T)
```

27.8 Compressione non lineare

```
def compress_nonlinear(x):
    return np.round(x)
```

27.9 Policy compressa

```
def compressed_policy(z):
    return z
```

27.10 Calcolo regret strutturale

def structural_regret(theta, Sigma, B):

    x = generate_samples(theta, Sigma, 10000)

    z = compress_linear(x, B)

    a_bar = np.mean(z, axis=0)

    full_opt = optimal_action(theta)

    r_full = empirical_risk(theta, full_opt, Sigma)

    r_comp = empirical_risk(theta, a_bar, Sigma)

    return r_comp - r_full

27.11 Verifica collasso

Per theta1, theta2 tali che:

B theta1 = B theta2

ma theta1 ≠ theta2,

verificare:

structural_regret(theta1) > 0
structural_regret(theta2) > 0

27.12 Parametrizzazione generale

Config JSON:


```
{
  "d_theta": 5,
  "m_action": 5,
  "compression_rank": 3,
  "loss": "quadratic",
  "samples": 10000,
  "seed": 42
}
```


27.13 Output atteso

Per ogni esperimento:

- regret_min
- regret_mean
- regret_max
- delta_theta
- rank_compression

27.14 Validazione teorica

Confrontare:

regret_empirico

con bound teorico:

(mu/8)||theta1 - theta2||^2

Deve rispettare disuguaglianza inferiore.

27.15 Conclusione Capitolo 27

Abbiamo definito infrastruttura base per simulazione strutturale.

È:

- generica
- modulare
- estendibile
- replicabile

CAPITOLO 28

ROBUSTEZZA SU GRIGLIA PARAMETRI

28.1 Obiettivo

Verificare che il fenomeno di incompletezza:

- non sia locale
- non dipenda da scelta specifica di theta
- non dipenda da singolo livello di compressione
- non dipenda da singolo regime di varianza

Costruiamo una **mappa strutturale del regret** nello spazio dei parametri.

28.2 Spazio parametri esplorato

Variabili da esplorare:

1. d_theta ∈ {2,5,10}
2. rank_compression ∈ {1,...,d_theta-1}
3. sigma ∈ {0.5,1,2}
4. distanza ||theta1 - theta2||
5. tipo loss ∈ {quadratic, logistic}
6. tipo compressione ∈ {linear, nonlinear}

28.3 Costruzione griglia

Definiamo griglia parametrica:

Theta_pairs =
{ (theta1,theta2) :
||theta1 - theta2|| = delta
e B theta1 = B theta2 }

Generazione automatica:

```
def generate_collapsed_pair(d, rank, delta):
    B = np.random.randn(rank, d)
    U, S, Vt = np.linalg.svd(B)
    nullspace = Vt[rank:].T
    v = nullspace[:,0]
    v = v / np.linalg.norm(v)
    theta1 = np.random.randn(d)
    theta2 = theta1 + delta * v
    return theta1, theta2, B
```

28.4 Esperimento grid search

results = []

for d in [2,5,10]:

    for rank in range(1,d):

        for delta in [0.5,1,2]:

            theta1, theta2, B = generate_collapsed_pair(d, rank, delta)

            r1 = structural_regret(theta1, np.eye(d), B)

            r2 = structural_regret(theta2, np.eye(d), B)

            results.append((d,rank,delta,r1,r2))

28.5 Analisi attesa

Per ogni configurazione con rank < d:

regret > 0

Crescita attesa:

regret ∝ delta^2

28.6 Mappa strutturale

Possiamo costruire heatmap:

- asse x: rank/d
- asse y: delta
- colore: regret medio

Atteso:

regret cresce quando:

- delta aumenta
- rank diminuisce

28.7 Stabilità rispetto a sigma

Aumentando sigma:

Bound teorico indipendente da sigma nel caso quadratico puro.

Verifica empirica:

regret rimane positivo.

28.8 Robustezza non-lineare

Compressione non lineare:

```
def compress_round(x):
    return np.round(x,1)
```

Ripetere grid.

Fenomeno deve persistere.

28.9 Analisi statistica

Calcolare:

- media regret
- deviazione standard
- intervallo di confidenza 95%

Verificare che zero non sia incluso.

28.10 Interpretazione

Se fenomeno stabile su tutta la griglia,
allora:

non è artefatto locale, ma proprietà strutturale robusta.

28.11 Estensione a d elevato

Test:

d = 50

rank = 5

Fenomeno deve intensificarsi.

Dimensione invisibile cresce → maggiore spazio collasso.

28.12 Conclusione Capitolo 28

La simulazione su griglia dimostra:

- stabilità
- generalità
- scalabilità
- indipendenza da scelta specifica

CAPITOLO 29

DISTRIBUTION SHIFT E LIMITE STRUTTURALE

29.1 Obiettivo

Mostrare che il limite strutturale:

- non scompare sotto shift distribuzionale,
- può anzi amplificarsi,
- è indipendente dalla stabilità train/test.

Consideriamo tre tipi di shift:

1. Covariate shift
2. Parameter drift
3. Support shift

29.2 Setup generale

Supponiamo:

Train distribution: P_theta

Test distribution: Q_theta

Compressione:

C(X)

Decisione basata su C.

Rischio test:

R_test(theta,a) =
E_{Q_theta}[ell(theta,a,X)]

29.3 Covariate shift

Caso:

Q_theta(X) = w(X) P_theta(X)

con w(X) funzione di densità relativa.

Se collasso avviene sotto P_theta, allora C#P_theta1 = C#P_theta2.

Se C#Q_theta1 = C#Q_theta2 rimane indistinguibilità.

Regret test resta ≥ epsilon.

29.4 Shift parametrico

Theta_train ≠ Theta_test.

Supponiamo:

theta_train1, theta_train2 collassano.

Nel test:

theta_test1 = theta_train1 + v

theta_test2 = theta_train2 + v

Se compressione lineare:

B(theta + v) = B theta + B v.

Se B v = 0,

collasso persiste.

29.5 Support shift

Distribuzioni con stesso Z ma differente supporto in X.

Esempio:

P1 concentrata su X1

P2 concentrata su X2

C(X1) = C(X2)

Anche se supporti non si sovrappongono, collasso informazionale resta.

29.6 Teorema di stabilità sotto shift

TEOREMA 29.1

Se compressione induce collasso tra theta1 e theta2 e shift preserva uguaglianza delle leggi compresse, allora regret test ≥ epsilon.

29.7 Caso di amplificazione

Se shift aumenta separazione tra superfici di rischio, allora:

epsilon_test ≥ epsilon_train.

Il limite può crescere.

29.8 Simulazione distribution shift

Estensione codice:

```
def shifted_samples(theta, Sigma, shift_vector, n):
    return np.random.multivariate_normal(theta + shift_vector, Sigma, n)
```

Calcolare regret su distribuzione shiftata.

29.9 Robustezza empirica

Ripetere grid search con:

shift_vector ∈ {0, small, medium}

Verificare:

regret non converge a zero.

29.10 Interpretazione teorica

Il limite è proprietà della rappresentazione, non della stabilità della distribuzione.

Anche con perfect generalization, collasso strutturale non si elimina.

29.11 Collegamento con robustezza (Cap. 23)

Versione robusta già garantisce stabilità sotto insiemi di distribuzioni.

Distribution shift è caso particolare.

29.12 Conclusione Capitolo 29

Il limite strutturale:

- non è legato al training set
- non è eliminabile con più dati
- non è eliminabile con domain adaptation

È vincolo intrinseco alla compressione.

CAPITOLO 30

PROTOCOLLO PREREGISTRATO COMPLETO

30.1 Obiettivo

Stabilire un protocollo che:

- definisce a priori tutte le ipotesi
- fissa parametri prima dell'esecuzione
- impedisce tuning post-hoc
- consente replica indipendente
- rende i risultati verificabili con hash

30.2 Ipotesi preregistrate

H1 — Incompletezza strutturale

Se compressione C collassa stati decisionalmente distinti, allora regret minimo empirico > 0.

H2 — Crescita quadratica

In caso strongly convex, regret ≈ c ||theta1 − theta2||^2.

H3 — Stabilità su griglia

Su griglia parametri con rank < d, regret medio > 0 con IC95% escluso 0.

H4 — Stabilità sotto shift

Distribution shift che preserva collasso non riduce regret a zero.

30.3 Parametri fissati prima dell'esecuzione

File config.json preregistrato:

```
{
  "d_theta": [2,5,10],
  "rank": "range(1,d_theta)",
  "delta": [0.5,1,2],
  "sigma": [0.5,1,2],
  "samples": 10000,
  "repetitions": 50,
  "seed": 42,
  "loss": "quadratic",
  "compression": "linear"
}
```

Non modificabile dopo registrazione.

30.4 Metriche dichiarate

1. regret_min
2. regret_mean
3. regret_max
4. std_regret
5. IC95

Calcolo IC:

IC95 = mean ± 1.96 * std / sqrt(n_repetitions)

30.5 Criterio di successo

Esperimento conferma teoria se:

Per ogni configurazione con rank < d:

regret_mean > 0
IC95_lower > 0.

30.6 Criterio di falsificazione

Teoria falsificata se esiste configurazione con:

- collasso strutturale verificato
- regret_mean ≈ 0
- IC95 include 0

- con n grande

30.7 Pipeline di esecuzione

Ordine obbligatorio:

1. Caricamento config
2. Generazione griglia
3. Generazione theta1,theta2 collassati
4. Calcolo regret
5. Salvataggio CSV
6. Calcolo summary
7. Salvataggio manifest SHA256

30.8 Output obbligatori

Files:

results_raw.csv

summary.csv

config.json

manifest.sha256

30.9 Manifest SHA256

Generazione:

sha256sum results_raw.csv > manifest.sha256

Garantisce integrità.

30.10 Blocco modifiche

Regola:

Nessun parametro può essere cambiato dopo generazione manifest.

Qualsiasi modifica genera nuovo hash.

30.11 Audit indipendente

Un revisore deve poter:

1. Scaricare repository
2. Eseguire script
3. Confrontare hash
4. Riprodurre IC

Se risultati divergono → problema.

30.12 Neutralità dell'autore

Il protocollo è:

- deterministico (seed fissato)
- trasparente
- non adattivo
- non retrofittato

30.13 Conclusione Capitolo 30

Il programma empirico:

- è falsificabile
- è preregistrato
- è replicabile
- è auditabile

Non è costruito ex post.

CAPITOLO 31

DATASET REALE E PIPELINE APPLICATIVA

31.1 Obiettivo

Verificare che il limite strutturale:

- non sia un artefatto simulativo
- emerga in dataset reali
- sia misurabile in pipeline concrete

31.2 Schema generale applicativo

Per ogni dataset reale:

1. Definire parametro implicito theta
2. Definire decisione target
3. Definire compressione C (feature reduction)
4. Calcolare azione ottima full-feature
5. Calcolare azione ottima compressa
6. Misurare regret empirico

31.3 Dataset candidati

Categorie generali:

- Regressione medica (es. predizione outcome continuo)
- Classificazione clinica
- Dataset finanziari
- Dataset controllo dinamico

La teoria non dipende dal dominio specifico.

31.4 Pipeline generale

Step 1 — Dataset

Caricamento:

import pandas as pd

data = pd.read_csv("dataset.csv")

Separazione:

X_full = feature complete
y = target

Step 2 — Modello completo

Esempio regressione lineare:

```
from sklearn.linear_model import LinearRegression


model_full = LinearRegression()

model_full.fit(X_full, y)

pred_full = model_full.predict(X_full)
```

Loss completa:

```
MSE_full = mean((pred_full - y)^2)
```

Step 3 — Compressione

Riduzione dimensionale:

- PCA
- Feature selection
- Projection B

Esempio PCA:

```
from sklearn.decomposition import PCA


pca = PCA(n_components=k)

X_comp = pca.fit_transform(X_full)
```

Step 4 — Modello compresso

model_comp = LinearRegression()

model_comp.fit(X_comp, y)

pred_comp = model_comp.predict(X_comp)

Loss compressa:

MSE_comp = mean((pred_comp - y)^2)

31.5 Definizione regret empirico

Regret_emp = MSE_comp − MSE_full

Se compressione elimina direzione decisionale rilevante,
Regret_emp > 0.

31.6 Verifica collasso decisionale

Identificare:

- Componenti principali con varianza alta ma irrilevanti
- Componenti con varianza bassa ma decisionali

Se PCA elimina componente decisionalmente rilevante, si verifica collasso.

31.7 Test controllato

Procedura:

1. Stimare coefficiente full-model
2. Identificare direzione rilevante
3. Costruire compressione che la elimina
4. Misurare aumento loss

31.8 Caso classificazione logistica

Full:

from sklearn.linear_model import LogisticRegression

Regret definito come:

Regret = LogLoss_comp − LogLoss_full

31.9 Cross-validation

Usare K-fold CV:

- evitare overfitting
- stimare media regret
- calcolare IC95

31.10 Distribution shift reale

Separare:

Train: anni 1–3

Test: anno 4

Verificare che regret persista sotto shift temporale.

31.11 Robustezza feature selection

Testare:

- eliminazione casuale feature
- eliminazione feature informative
- random projection

Regret deve crescere quando compressione elimina direzione rilevante.

31.12 Output preregistrato

Per ogni dataset:

- k componenti
- regret medio
- deviazione
- IC95
- hash risultati

31.13 Interpretazione

Se su dataset reali:

regret sistematicamente > 0
quando compressione elimina informazione decisionale, allora teoria validata empiricamente.

31.14 Limiti applicativi

Non si dimostra:

- che ogni compressione sia cattiva
- che ogni riduzione dimensionale fallisca

Si dimostra:

Se compressione collassa stati decisionalmente distinti, allora regret strutturale inevitabile.

31.15 Conclusione Capitolo 31

La teoria è:

- empiricamente testabile
- applicabile a contesti reali
- non limitata a modelli sintetici

CAPITOLO 32

CRITERI DI FALSIFICAZIONE

32.1 Principio generale

La teoria afferma:

Se una compressione collassa stati decisionalmente distinti, allora esiste regret strutturale positivo inevitabile.

La teoria è falsa se esiste controesempio valido sotto le ipotesi dichiarate.

32.2 Falsificazione strutturale (statica)

Controesempio valido se esistono:

theta1 ≠ theta2 tali che

1. C#(theta1) = C#(theta2)
2. a*(theta1) ≠ a*(theta2)
3. Esiste policy pi basata su C tale che
   regret(theta1;pi) = 0
   regret(theta2;pi) = 0

Se ciò accade, il teorema principale è falso.

32.3 Falsificazione minimax

Se esiste collasso e tuttavia:

inf_pi sup_theta regret(theta;pi) = 0

allora la teoria è falsa.

32.4 Falsificazione differenziale

Se esiste theta e direzione v tali che:

DC(theta)[v] = 0
D a*(theta)[v] ≠ 0

ma per ogni t sufficientemente piccolo:

regret(theta + t v) = 0,

allora la formulazione differenziale è falsa.

32.5 Falsificazione informazionale

Se esistono P1, P2 tali che:

C#P1 = C#P2
a*(P1) ≠ a*(P2)

ma regret minimo = 0,

allora connessione tra perdita informazione e regret è falsa.

32.6 Falsificazione empirica

Nel protocollo preregistrato:

Se per configurazione con collasso verificato:

regret_mean ≈ 0
IC95 include 0
con n grande, allora teoria empiricamente falsificata.

32.7 Falsificazione robusta

Se in regime robusto:

P_theta1^Z = P_theta2^Z
a_*rob(theta1)* ≠ *a*_rob(theta2)

ma regret robusto minimo = 0, allora estensione robusta falsa.

32.8 Cosa NON falsifica la teoria

Non falsifica:

- Assenza di collasso
- Compressione che preserva sufficienza decisionale
- Regret piccolo ma > 0
- Prior degeneri

La teoria non afferma incompletezza universale, ma incompletezza sotto collasso.

32.9 Condizioni necessarie

La teoria richiede:

1. Esistenza di azione ottima ben definita
2. Separazione locale del rischio
3. Collasso decisionale reale
4. Policy dipendente solo da compressione

Se una di queste manca, il teorema non si applica.

32.10 Falsificazione concettuale

Se si dimostra che:

Esiste trasformazione T tale che C collassa stati ma esiste funzione g(Z) che ricostruisce informazione decisionale perduta, allora C non era vera compressione nel senso definito.

La definizione stessa va rivista.

32.11 Stato epistemologico

La teoria è:

- falsificabile
- non tautologica
- non dipendente da parametri nascosti
- strutturalmente controllabile

32.12 Conclusione Parte VIII

Abbiamo:

- costruito simulazioni
- esplorato griglia
- testato shift
- definito preregistrazione
- definito criteri di falsificazione

Il programma empirico è completo.

PARTE IX

FORMALIZZAZIONE MACHINE-CHECKED

CAPITOLO 33

DEFINIZIONI LEAN — STRUTTURA ASSIOMATICA FORMALE

33.1 Obiettivo

Tradurre la teoria in:

- oggetti tipizzati
- definizioni rigorose
- proprietà verificabili
- prerequisiti espliciti

Ambiente target: **Lean 4 + Mathlib**

33.2 Spazi primitivi

Definizione dei tipi fondamentali:

universe u

variable {Θ : Type u}

variable {X : Type u}

variable {A : Type u}

variable {Z : Type u}

33.3 Struttura probabilistica

Assumiamo:

variable (P : Θ → ProbabilityMeasure X)

Rischio atteso:

def Risk (θ : Θ) (a : A) : ℝ :=

∫ x, loss θ a x ∂(P θ)

33.4 Loss

variable (loss : Θ → A → X → ℝ)

Ipotesi minime:

- integrabilità
- continuità (eventuale)
- convessità in a (se richiesta)

33.5 Azione ottima

Se A finito:

def OptimalAction (θ : Θ) : A :=

argmin (fun a => Risk loss P θ a)

Se A continuo:

richiede struttura topologica e compattezza.

33.6 Compressione

variable (C : X → Z)

Pushforward:

def PushForward (θ : Θ) : ProbabilityMeasure Z :=

(P θ).map C

33.7 Definizione di collasso

def Collapsed (θ₁ θ₂ : Θ) : Prop :=

PushForward P C θ₁ = PushForward P C θ₂

∧ OptimalAction loss P θ₁ ≠ OptimalAction loss P θ₂

33.8 Policy compressa

variable (π : Z → A)

Regret:

def Regret (θ : Θ) (π : Z → A) : ℝ :=

Risk loss P θ (π (C_sample θ))

- Risk loss P θ (OptimalAction loss P θ)

Nota: C_sample θ formalizza osservazione media.

33.9 Ipotesi di separazione

def Separated (θ : Θ) : Prop :=

∀ a ≠ OptimalAction loss P θ,

Risk loss P θ a > Risk loss P θ (OptimalAction loss P θ)

33.10 Teorema di incompletezza (forma esistenziale)

theorem structural_incompleteness

(h : ∃ θ₁ θ₂, Collapsed loss P C θ₁ θ₂)

(sep : ∀ θ, Separated loss P θ) :

∃ ε > 0,

∀ π : Z → A,

∃ θ,

Regret loss P C θ π ≥ ε

33.11 Struttura delle dipendenze

Dipendenze richieste:

- MeasureTheory
- Integration

- Topology (se A compatto)
- FiniteType (se A finito)
- Convexity (se A continuo)

33.12 Versione minimax

def MinimaxRegret : ℝ :=

inf (fun π => sup (fun θ => Regret loss P C θ π))

Teorema:

theorem minimax_positive

(h : ∃ θ₁ θ₂, Collapsed loss P C θ₁ θ₂)

(sep : ...)

: MinimaxRegret loss P C > 0

33.13 Estensione differenziale

Richiede:

- Θ topological space
- Differentiable functions
- Rank deficiency

Definizione:

```
def RankDeficient (θ : Θ) : Prop :=
  rank (differential C θ) < dim_dec
```

33.14 Struttura modulare repository

```
DecisionTheory/
  Core/
    Spaces.lean
    Risk.lean
    Compression.lean
    Collapse.lean
  Minimax/
    Regret.lean
    Minimax.lean
  Information/
    TV.lean
    KL.lean
  Differential/
    RankCollapse.lean
  Functional/
```

Projection.lean

33.15 Obiettivo machine-check

Un repository completo deve:

- compilare senza admit
- avere CI
- avere hash
- avere tag versionato
- avere SWHID

33.16 Stato attuale

La formalizzazione è:

- strutturalmente coerente
- pronta per implementazione completa
- modulare

CAPITOLO 34

TEOREMA MINIMAX FORMALIZZATO

34.1 Obiettivo

Formalizzare in Lean il risultato centrale:

Se esiste collasso decisionale, allora il regret minimax è strettamente positivo.

Forma matematica:

$\inf_{\pi} \sup_{\theta} \mathrm{Regret}(\theta;\pi) > 0$

34.2 Setup formale richiesto

Per avere prova completamente formalizzabile, assumiamo:

1. Θ finito oppure compatto metrico
2. A finito oppure compatto convesso
3. Risk continua
4. Separazione stretta del minimo

Per prima formalizzazione machine-checked, caso più semplice:

Θ finito
A finito

34.3 Versione discreta Lean-ready

Ipotesi

variable [Fintype Θ]

variable [Fintype A]

Separazione:

axiom separation :

∀ θ a, a ≠ OptimalAction loss P θ →

Risk loss P θ a >

Risk loss P θ (OptimalAction loss P θ)

Collasso:

axiom collapse :

∃ θ₁ θ₂,

PushForward P C θ₁ = PushForward P C θ₂ ∧

OptimalAction loss P θ₁ ≠ OptimalAction loss P θ₂

34.4 Costruzione epsilon

Poiché Θ e A finiti,
l'insieme dei valori:

{ Risk θ a - Risk θ a*(θ) | θ,a }

è finito.

Sia:

ε = min_{θ,a≠a*(θ)}
(Risk θ a - Risk θ a*(θ))

Per separazione,
ε > 0.

Formalizzazione:

def epsilon : ℝ :=

  Finset.inf' ...

Lemma:

lemma epsilon_pos : epsilon loss P > 0

34.5 Passo centrale

Dato collasso θ_1, θ_2,
per ogni policy π:

π(C_sample θ_1) = π(C_sample θ_2) = a_bar

Almeno uno tra:

a_bar ≠ a*(θ_1)
oppure
a_bar ≠ a*(θ_2)

Quindi:

Regret ≥ ε

Formalizzabile con casi su uguaglianza.

34.6 Teorema minimax discreto

theorem minimax_positive :

∃ ε > 0,

∀ π : Z → A,

∃ θ,

Regret loss P C θ π ≥ ε

Proof outline:

1. Estrarre θ_1, θ_2 dal collasso
2. Definire ε tramite minimo finito
3. Fissare π arbitrario
4. Dimostrare per casi che uno dei due θ produce regret ≥ ε

QED.

34.7 Estensione a minimax esplicito

Definiamo:

def MinimaxRegret : ℝ :=

sInf { r | ∃ π, ∀ θ, Regret θ π ≤ r }

Mostrare:

theorem minimax_strict :

MinimaxRegret loss P C ≥ epsilon loss P

Segue:

MinimaxRegret > 0

34.8 Versione compatta continua (sketch)

Richiede:

- Θ compatto
- A compatto
- Risk continua
- separazione stretta

Struttura:

1. Usare compattezza per ottenere minimo positivo locale
2. Usare collasso per riduzione a caso binario
3. Applicare argomento discreto su coppia

Formalizzazione richiede:

- TopologicalSpace
- Continuous
- IsCompact
- Attainable minimum lemma

34.9 Dipendenze Mathlib

Richieste:

- MeasureTheory.Integral
- Topology.Compactness
- Analysis.Convex.StronglyConvex
- Order.CompleteLattice

34.10 Completezza proof-grade

Versione finita: 100% formalizzabile senza assiomi aggiuntivi.

Versione continua: richiede formalizzazione tecnica più lunga ma standard.

34.11 Stato logico

Il teorema:

- non usa argomenti informali
- non dipende da euristiche
- è riducibile a combinatoria finita nel caso discreto
- è estendibile al continuo tramite compattezza

Questo conclude la formalizzazione del teorema minimax.

CAPITOLO 35

BOUND TV FORMALIZZATO

35.1 Obiettivo

Dimostrare formalmente:

Se la compressione riduce la distanza TV tra due distribuzioni, e le azioni ottime differiscono, allora esiste bound inferiore positivo sul regret.

35.2 Definizione TV

Per due misure P,Q su spazio misurabile:

TV(P,Q) =
sup_{A misurabile} |P(A) − Q(A)|

In Lean (Mathlib):

def TV (μ ν : Measure X) : ℝ :=

(1/2) * ∫ x, |dμ/dλ - dν/dλ| dλ

(dove λ misura dominante)

Oppure usare formulazione equivalente già presente in Mathlib.

35.3 Proprietà fondamentale

Lemma standard (Data Processing Inequality per TV):

Se $C : X \to Z$ misurabile,

allora:

$TV(P.map\ C, Q.map\ C) \leq TV(P,Q)$.

Formalizzabile in Lean come:

lemma tv_data_processing :

TV (P.map C) (Q.map C) ≤ TV P Q

35.4 Setup decisionale binario

Caso più semplice formalizzabile:

$\Theta = \{\theta_1, \theta_2\}$

$A = \{a_1, a_2\}$

Supponiamo:

$a^*(\theta_1) = a_1$

$a^*(\theta_2) = a_2$

$a_1 \neq a_2$

35.5 Connessione con testing

Qualsiasi policy π basata su Z

induce test $\varphi(Z)$ che sceglie tra a_1 e a_2.

Errore minimo di test è legato a TV:

$P_error \geq (1 - TV(P_Z_1, P_Z_2)) / 2$

Lemma Lean-ready:

lemma testing_error_bound :

$P_error \geq (1 - TV\ (P_1.map\ C)\ (P_2.map\ C)) / 2$

35.6 Connessione con regret

Supponiamo loss 0-1:

$ell(\theta,a) =$

0 se $a = a^*(\theta)$

1 altrimenti

Allora:

$Regret = P_error$

Segue:

$Regret \geq (1 - TV_Z) / 2$

35.7 Collasso totale

Se:

$P_1.\text{map } C = P_2.\text{map } C$

allora:

$TV_Z = 0$

Quindi:

$\text{Regret} \geq 1/2$

Bound esplicito positivo.

35.8 Formalizzazione Lean del caso binario

Struttura:

```
theorem tv_regret_lower_bound
  (h_opt : OptimalAction θ₁ ≠ OptimalAction θ₂)
  (h_loss01 : loss = zero_one_loss)
  :
  ∀ π,
    Regret θ₁ π + Regret θ₂ π
      ≥ 1 - TV (P θ₁).map C (P θ₂).map C
```

Caso collasso:

corollary collapse_implies_positive_regret :

PushForward P C θ_1 = PushForward P C θ_2 →

∀ π,

Regret θ_1 π + Regret θ_2 π ≥ 1

35.9 Estensione multi-classe

Per Θ finito, regret legato a testing multi-ipotesi.

Lower bound via:

minimax risk ≥ 1 − max_pair TV_Z.

Formalizzabile tramite riduzione binaria.

35.10 Connessione con teorema principale

Se:

TV_X(θ_1,θ_2) > 0
ma
TV_Z(θ_1,θ_2) = 0,

allora compressione elimina capacità di distinguere stati decisionalmente distinti.

Regret positivo inevitabile.

35.11 Struttura proof-grade

Dipendenze Lean:

- MeasureTheory
- L1 norm
- Radon–Nikodym
- Testing lemmas

Il caso binario 0-1 è completamente formalizzabile senza assiomi esterni.

35.12 Conclusione Capitolo 35

Abbiamo formalizzato:

- Data Processing per TV
- Connessione testing → regret
- Bound esplicito in caso collasso

Questo chiude la versione informazionale TV machine-checkable.

CAPITOLO 36

PIANO COQ ALTERNATIVO E INDIPENDENZA FORMALE

36.1 Obiettivo

Fornire un'architettura Coq che:

- riproduce integralmente la struttura assiomatica
- formalizza il teorema di incompletezza
- formalizza il bound TV
- consente verifica indipendente
- non dipende da Lean o Mathlib

Ambiente target:

Coq 8.18+
Librerie:

- Coquelicot (analisi reale)
- Mathematical Components (finite types)
- MathComp Analysis (misura)

36.2 Struttura dei moduli Coq

Proposta repository:

DecisionIncomplete/

Core/

Spaces.v

Risk.v

Compression.v

Collapse.v

Minimax/

Regret.v

MinimaxTheorem.v

Information/

TV.v

Testing.v

Differential/

RankCollapse.v

Functional/

Projection.v

36.3 Tipi fondamentali

In Coq:

Parameter Theta : Type.

Parameter X : Type.

Parameter A : Type.

Parameter Z : Type.

Misure:

Parameter P : Theta -> Measure X.

Loss:

Parameter loss : Theta -> A -> X -> R.

36.4 Rischio

Definition Risk (θ : Theta) (a : A) :=

  Integral (P θ) (fun x => loss θ a x).

36.5 Azione ottima

Caso finito:

Parameter OptimalAction : Theta -> A.

Con assioma di correttezza:

Axiom optimality :

  forall θ a,

    Risk θ (OptimalAction θ) <= Risk θ a.

36.6 Compressione

Parameter C : X -> Z.

Definition PushForward (θ : Theta) :=

  map_measure (P θ) C.

36.7 Definizione collasso

Definition Collapsed (θ1 θ2 : Theta) :=

PushForward θ1 = PushForward θ2 /\

OptimalAction θ1 <> OptimalAction θ2.

36.8 Teorema principale Coq

Theorem structural_incompleteness :

(exists θ1 θ2, Collapsed θ1 θ2) ->

exists ε > 0,

forall π : Z -> A,

exists θ,

Risk θ (π (C_sample θ))

- Risk θ (OptimalAction θ) >= ε.

Proof strategy:

1. Caso finito
2. Costruzione ε come minimo finito
3. Riduzione a coppia collassata

36.9 Formalizzazione TV in Coq

Definizione TV:

```
Definition TV (μ ν : Measure X) :=
  sup (fun A => Rabs (μ A - ν A)).
```

Data processing:

```
Theorem tv_data_processing :
  forall μ ν,
    TV (map_measure μ C) (map_measure ν C)
    <= TV μ ν.
```

Testing lemma:

```
Theorem testing_lower_bound :
  P_error >= (1 - TV_Z) / 2.
```

36.10 Differenziale in Coq

Con Coquelicot:

- Deriv
- Jacobian
- Rank

Definizione:

Definition RankDeficient θ :=

  rank (Jacobian C θ) < dim_dec.

Teorema locale:

Rank deficiency → collasso differenziale → regret positivo.

36.11 Confronto Lean vs Coq

Aspetto	Lean	Coq
Ecosistema measure	Mathlib	MathComp Analysis
Analisi reale	Integrata	Coquelicot
Automazione	tactic-heavy	più esplicita
Dipendenza	Mathlib	più modulare

La teoria è formalizzabile in entrambi.

36.12 Indipendenza epistemica

Formalizzazione in due proof assistant distinti implica:

- Indipendenza da bug di singolo sistema
- Riduzione rischio errori nascosti
- Robustezza logica

36.13 Stato proof-grade globale

Ora la teoria è:

- Assiomatizzata
- Formalizzabile in Lean
- Formalizzabile in Coq
- Collegata a measure theory
- Collegata a convex analysis
- Collegata a information theory

36.14 Conclusione Parte IX

La teoria è:

- computazionalmente verificabile
- indipendente da implementazione
- priva di ambiguità semantiche
- auditabile

PARTE X

APPENDICI TECNICHE COMPLETE

APPENDICE A

DIMOSTRAZIONI INTEGRALI DETTAGLIATE

A.1 Dimostrazione completa del Teorema di Incompletezza (caso discreto finito)

Setup

Sia:

- Θ finito
- A finito
- Loss integrabile
- $R(\theta,a)$ rischio atteso

Definiamo:

$a^*(\theta) = \text{argmin}_a R(\theta,a)$

Ipotesi:

1. Esistono θ_1, θ_2 tali che
   $C\#(\theta_1) = C\#(\theta_2)$
2. $a^*(\theta_1) \neq a^*(\theta_2)$
3. Separazione stretta:
   $\forall \theta, a \neq a^*(\theta)$
   $R(\theta,a) > R(\theta,a^*(\theta))$

A.1.1 Costruzione esplicita di ε

Definiamo insieme:

$S = \{ R(\theta,a) - R(\theta,a^*(\theta)) :$

$\theta \in \Theta,$

$a \in A,$

$a \neq a^*(\theta) \}$

Poiché Θ e A finiti, S è finito.

Poiché separazione stretta, ogni elemento di $S > 0$.

Definiamo:

$\varepsilon = \min S$

Allora:

$\varepsilon > 0.$

A.1.2 Regret per policy arbitraria

Sia $\pi : Z \rightarrow A$.

Poiché $C\#(\theta_1) = C\#(\theta_2)$,

$\pi(C\#(\theta_1)) = \pi(C\#(\theta_2)) = a_bar.$

Due casi:

Caso 1:

$a_bar \neq a^*(\theta_1)$

Allora:

$R(\theta_1, a_bar) - R(\theta_1, a^*(\theta_1)) \geq \varepsilon$.

Caso 2:

$a_bar = a^*(\theta_1)$.

Ma poiché $a^*(\theta_1) \neq a^*(\theta_2)$,

$a_bar \neq a^*(\theta_2)$.

Allora:

$R(\theta_2, a_bar) - R(\theta_2, a^*(\theta_2)) \geq \varepsilon$.

In entrambi i casi:

$\exists \theta \in \{\theta_1, \theta_2\}$ tale che regret $\geq \varepsilon$.

QED.

A.2 Dimostrazione completa bound strongly convex

Assumiamo:

$R(\theta, a)$ è μ-strongly convex in a.

Definizione:

$R(\theta, a) \geq$
$R(\theta, a^*(\theta))$

- $(\mu/2)\,||a - a^*(\theta)||^2$.

A.2.1 Caso binario continuo

Siano:

$a_1 = a^*(\theta_1)$

$a_2 = a^*(\theta_2)$

$\delta = ||a_1 - a_2||$.

Per qualsiasi a_bar:

max(

$||a_bar - a_1||^2$,

$||a_bar - a_2||^2$

)

$\geq (\delta^2 / 4)$.

(Minimo nel punto medio.)

Quindi:

max regret $\geq (\mu/2)(\delta^2/4) = (\mu/8)\delta^2$.

QED.

A.3 Dimostrazione differenziale locale completa

Assumiamo:

$\Theta \subset R^d$ aperto

$a^*(\theta)$ differenziabile

C differenziabile

Supponiamo:

$DC(\theta)[v] = 0$

$D\ a^*(\theta)[v] \neq 0$

A.3.1 Espansione di Taylor

Per piccolo t:

$a^*(\theta + t\ v)$

$= a^*(\theta)$

- $t\ D\ a^*(\theta)[v]$
- $O(t^2)$.

Quindi:

$|| a^*(\theta + t\ v) - a^*(\theta) ||$

$\geq c\ |t|$

per qualche $c > 0$.

A.3.2 Strong convexity

Regret $\geq (\mu/2)\ ||\Delta a||^2$

$\geq (\mu/2)(c^2\ t^2)$

$= O(t^2)$.

Per $t \neq 0$ piccolo:

regret > 0.

QED.

A.4 Dimostrazione completa bound TV

Caso binario con loss 0–1.

Errore minimo di test:

$P_error =$

$(1 - TV(P_Z_1, P_Z_2)) / 2.$

Se $P_Z_1 = P_Z_2$,

$TV = 0$

$\Rightarrow P_error = 1/2.$

Regret = P_error.

QED.

A.5 Dimostrazione data processing inequality per TV

Sia:

$TV(P,Q)$

$= \sup_A \, |P(A) - Q(A)|.$

Per ogni $B \subset Z$ misurabile:

$(P.map\ C)(B)$

$= P(C^{-1}(B)).$

Quindi:

| (P.map C)(B) − (Q.map C)(B) |

| $P(C^{-1}(B)) - Q(C^{-1}(B))$ |
≤ TV(P,Q).

Prendendo sup su B:

TV(P.map C, Q.map C)
≤ TV(P,Q).

QED.

A.6 Dimostrazione equivalenza funzionale L2

Sia:

T_C f = E[f|Z].

Se f non è Z-misurabile,
allora:

$||(I - T_C)f||_2 > 0$.

Allora esistono insiemi A,B tali che f differisce su punti con stesso Z.

Costruendo misure concentrate, si ottiene differenza di aspettativa.

⇒ collasso decisionale.

QED.

APPENDICE B

ALGEBRA FUNZIONALE COMPLETA

B.1 Spazio di Hilbert L2(P)

Sia (X, F, P) spazio di probabilità.

Definiamo:

L2(P) =
{ f : X → R misurabile |
∫ f(x)^2 dP(x) < ∞ }

Prodotto interno:

⟨f,g⟩ =
∫ f(x) g(x) dP(x)

Norma:

$||f||_2 = (\int f(x)^2 dP(x))^{(1/2)}$

Proprietà:

1. Spazio vettoriale
2. Completo
3. Hilbert separabile (se X standard Borel)

B.2 Sottospazio generato da compressione

Sia $C : X \to Z$.

Definiamo sigma-algebra:

$G = \sigma(C)$

Sottospazio:

$H_Z =$

$\{ f \in L2(P) :$

f è G-misurabile $\}$

Equivalentemente:

$f(x) = g(C(x))$

per qualche $g : Z \to R$.

H_Z è chiuso in $L2(P)$.

B.3 Proiezione ortogonale

Teorema classico:

Per ogni $f \in L2(P)$

esiste unico $h \in H_Z$ tale che:

$|| f - h ||_2$ minimo.

Questo h è:

$T_C f = E[f \mid Z]$.

Proprietà:

1. T_C è lineare
2. T_C è autoaggiunto
3. T_C^2 = T_C
4. ||T_C|| ≤ 1

B.4 Decomposizione ortogonale

Per ogni f:

f = T_C f + (I − T_C) f

con:

⟨ T_C f , (I − T_C) f ⟩ = 0.

Interpretazione:

T_C f = componente visibile
(I − T_C) f = componente invisibile

B.5 Spazio decisionale funzionale

Per ogni θ, definiamo:

f_{θ,a}(x) = ell(θ,a,x)

Spazio generato:

S_θ = span { f_{θ,a} : a ∈ A }

Spazio globale:

$S = \text{closure}(\cup_\theta S_\theta)$

B.6 Collasso funzionale

Definizione:

Collasso funzionale se

$\exists\ \theta$ tale che

$(I - T_C)\, f_{\{\theta,a^*(\theta)\}} \neq 0.$

Cioè la loss ottima ha componente invisibile.

B.7 Teorema funzionale dettagliato

Teorema B.1

Se per qualche θ

$||(I - T_C)\, f_{\{\theta,a^*(\theta)\}}||_2 > 0,$

allora esistono P1,P2 tali che:

$C\#P1 = C\#P2$

ma

$R(\theta,a^*(\theta);P1) \neq R(\theta,a^*(\theta);P2).$

Dimostrazione completa

1. Se norma > 0,
   esiste insieme A con P(A) > 0
   su cui f − T_C f ≠ 0.

2. Poiché T_C f è funzione di Z, esistono x_1,x_2 con:

$C(x_1)=C(x_2)$

ma

$f(x_1) \neq f(x_2)$.

3. Costruiamo misure:

$P1 = \delta_{x_1}$

$P2 = \delta_{x_2}$

4. C#P1 = C#P2

5. R differisce.

QED.

B.8 Caso compressione lineare

Sia:

C(x) = Bx

Spazio H_Z = funzioni dipendenti da Bx.

Se esiste direzione v con:

Bv = 0

e f varia lungo v, allora componente invisibile ≠ 0.

B.9 Caso non lineare

Anche se C non lineare, σ(C) definisce G.

T_C resta proiezione condizionale.

Il risultato non dipende da linearità.

B.10 Equivalenza con rank deficiency

Se Θ ⊂ R^d e C differenziabile, Rank(DC(θ)) < dim_dec

⇒ esistono direzioni v invisibili

⇒ esiste componente funzionale invisibile

⇒ collasso funzionale.

B.11 Connessione con informazione

Varianza decomposizione:

Var(f) =
Var(E[f|Z]) +
E[Var(f|Z)]

Il secondo termine è:

$|| (I - T_C) f ||_2^2$.

Perdita di varianza condizionale
→ perdita di informazione decisionale.

B.12 Interpretazione finale

L'incompletezza è equivalente a:

Perdita di componente ortogonale nello spazio L2 delle funzioni di perdita.

È fatto puramente operatoriale.

APPENDICE C

DERIVAZIONI INTEGRALI COMPLETE

(KL, Pinsker, Fano, f-divergence)

C.1 Total Variation — Forma Integrale

Siano μ, ν misure di probabilità assolutamente continue rispetto a λ con densità p, q.

Definizione:

TV(μ,ν)

sup_A |μ(A) − ν(A)|

Forma equivalente:

TV(μ,ν)

(1/2) ∫ |p(x) − q(x)| dλ(x)

Dimostrazione equivalenza

Definiamo:

A* = { x : p(x) ≥ q(x) }.

Allora:

∫_{A*} (p − q) dλ

(1/2) ∫ |p − q| dλ.

E si verifica che A* realizza il sup.

QED.

C.2 Divergenza di Kullback–Leibler

Definizione:

KL(P||Q)

$\int p(x) \log(p(x)/q(x))\, d\lambda(x)$

Proprietà:

1. KL ≥ 0
2. KL = 0 iff P=Q
3. Non simmetrica

C.3 Disuguaglianza di Pinsker

Enunciato:

TV(P,Q)
≤
sqrt((1/2) KL(P||Q))

Dimostrazione (sketch completo)

Usiamo:

log u ≥ 1 − 1/u.

Definiamo:

$f(u) = u \log u - u + 1$.

$f(u) \geq (u-1)^2 / 2$ per u vicino a 1.

Applicando disuguaglianza punto per punto:

$\int |p-q|$
$\leq$
sqrt(2 KL).

Quindi:

$TV \leq sqrt(KL/2)$.

QED.

C.4 Data Processing per KL

Se $C : X \to Z$ misurabile,

$KL(P.map\ C\ ||\ Q.map\ C)$
$\leq$
$KL(P||Q)$.

Dimostrazione:

Usa convexità di u log u e Jensen applicato alla trasformazione Markoviana indotta da C.

Formalmente:

P_Z(z)

$\int_{C^{-1}(z)} p(x)\, dx$.

Convessità implica riduzione divergenza.

QED.

C.5 Fano — Forma dettagliata

Caso binario $\Theta \in \{1,2\}$ con prior uniforme.

Errore minimo:

$P_e \geq$
$(1 - I(\Theta;Z) - \log 2)/\log 2.$

Poiché log 2 = 1 in base 2:

$P_e \geq 1 - I(\Theta;Z).$

Derivazione

I(Θ;Z)

$H(\Theta) - H(\Theta|Z)$

$H(\Theta)$=1 bit.

$H(\Theta|Z) \geq H(P_e)$

Usando monotonicità entropia binaria:

$P_e \geq 1 - I(\Theta;Z).$

C.6 Collegamento a regret

Per loss 0–1:

Regret = P_error.

Quindi:

Regret ≥ 1 − I(Θ;Z).

Se I=0 (collasso totale):

Regret ≥ 1.

C.7 f-divergence generale

Definizione:

D_f(P||Q)

∫ q(x) f(p(x)/q(x)) dλ(x)

con f convessa e f(1)=0.

TV e KL sono casi particolari.

C.8 Data processing per f-divergence

Per trasformazione Markoviana T:

D_f(P∘T || Q∘T)
≤
D_f(P||Q)

Dimostrazione via Jensen su kernel di transizione.

C.9 Bound generale regret–divergenza

In caso binario:

Regret ≥ g(D_f_Z)

dove g monotona decrescente in D_f_Z.

In particolare:

Se D_f_Z = 0

⇒ Regret positivo massimo.

C.10 Sintesi informazionale

Abbiamo dimostrato:

1. TV forma integrale
2. Pinsker
3. Data processing per KL
4. Fano
5. f-divergence generale
6. Collegamento testing → regret

Il limite strutturale è conseguenza diretta di:

Perdita di divergenza sotto compressione.

APPENDICE D

CODICE PYTHON COMPLETO (PACCHETTO REPLICABILE)

D.1 Struttura file

Struttura consigliata directory:

decision_incompleteness/

config.json

main.py

simulation.py

compression.py

loss.py

regret.py

grid.py

shift.py

utils.py

D.2 config.json

```
{
  "d_theta": 5,
  "compression_rank": 3,
  "delta_values": [0.5, 1.0, 2.0],
  "sigma_values": [0.5, 1.0, 2.0],
  "samples": 10000,
  "repetitions": 50,
  "seed": 42,
  "loss": "quadratic",
  "compression": "linear",
  "shift": [0.0, 0.5]
}
```

D.3 utils.py

```
import numpy as np

import json


def load_config(path="config.json"):

    with open(path, "r") as f:

        return json.load(f)


def set_seed(seed):

    np.random.seed(seed)


def confidence_interval(mean, std, n):

    return mean - 1.96*std/np.sqrt(n), mean + 1.96*std/np.sqrt(n)
```

D.4 loss.py

```
import numpy as np


def quadratic_loss(theta, a, x):
    return np.sum((a - theta)**2)


def logistic_loss(theta, a, x):
    y = (x.dot(theta) > 0).astype(int)
    eps = 1e-8
    return - (y*np.log(a+eps) + (1-y)*np.log(1-a+eps))
```

D.5 compression.py

```
import numpy as np


def linear_compression_matrix(d, rank):
    B = np.random.randn(rank, d)
    return B


def compress_linear(x, B):
    return x.dot(B.T)


def compress_round(x, decimals=1):
    return np.round(x, decimals)
```

D.6 simulation.py

```
import numpy as np


def generate_samples(theta, Sigma, n):
    return np.random.multivariate_normal(theta, Sigma, n)


def empirical_risk(theta, a, Sigma, n, loss_fn):
    x = generate_samples(theta, Sigma, n)
    losses = [loss_fn(theta, a, xi) for xi in x]
    return np.mean(losses)
```

D.7 regret.py

```
import numpy as np


def optimal_action(theta):
    return theta


def structural_regret(theta, Sigma, B, loss_fn, n):
    x = np.random.multivariate_normal(theta, Sigma, n)
    z = x.dot(B.T)
    a_bar = np.mean(z, axis=0)
    a_full = optimal_action(theta)
    r_full = np.mean([loss_fn(theta, a_full, xi) for xi in x])
    r_comp = np.mean([loss_fn(theta, a_bar, xi) for xi in x])
    return r_comp - r_full
```

D.8 grid.py

```
import numpy as np

from regret import structural_regret

from compression import linear_compression_matrix

from simulation import generate_samples

from utils import confidence_interval


def generate_collapsed_pair(d, rank, delta):

    B = linear_compression_matrix(d, rank)

    U, S, Vt = np.linalg.svd(B)

    nullspace = Vt[rank:].T

    v = nullspace[:,0]

    v = v / np.linalg.norm(v)

    theta1 = np.random.randn(d)

    theta2 = theta1 + delta * v

    return theta1, theta2, B


def run_grid(config, loss_fn):

    results = []

    d = config["d_theta"]
```

```
n = config["samples"]
reps = config["repetitions"]
for rank in range(1, d):
    for delta in config["delta_values"]:
        regrets = []
        for _ in range(reps):
            theta1, theta2, B = generate_collapsed_pair(d, rank, delta)
            Sigma = np.eye(d)
            r1 = structural_regret(theta1, Sigma, B, loss_fn, n)
            r2 = structural_regret(theta2, Sigma, B, loss_fn, n)
            regrets.append((r1+r2)/2)
        mean = np.mean(regrets)
        std = np.std(regrets)
        ci_low, ci_high = confidence_interval(mean, std, reps)
        results.append({
            "rank": rank,
            "delta": delta,
            "mean_regret": mean,
            "std": std,
            "ci_low": ci_low,
```

```
            "ci_high": ci_high
        })
    return results
```

D.9 shift.py

```
import numpy as np


def shifted_samples(theta, Sigma, shift_vector, n):
    return np.random.multivariate_normal(theta + shift_vector, Sigma, n)
```

D.10 main.py

```
import json

from utils import load_config, set_seed

from loss import quadratic_loss

from grid import run_grid


def main():

    config = load_config()

    set_seed(config["seed"])

    loss_fn = quadratic_loss

    results = run_grid(config, loss_fn)

    with open("results_raw.json", "w") as f:

        json.dump(results, f, indent=2)


if __name__ == "__main__":

    main()
```

D.11 Output atteso

File generato:

results_raw.json

Contiene:

- rank
- delta
- mean_regret
- std
- CI

D.12 Requisiti ambiente

Python 3.10+
numpy

Installazione:

pip install numpy

D.13 Riproducibilità completa

Procedura:

1. Copiare tutti file in directory
2. Salvare config.json
3. Eseguire:

python main.py

4. Verificare risultati

D.14 Interpretazione risultati

Con collasso:

mean_regret > 0
ci_low > 0

Se ci_low <= 0 su collasso verificato → potenziale falsificazione.

APPENDICE E

JSON CONFIG FORMALIZZATO

E.1 Struttura completa configurazione

File: config.json


```
{
 "experiment_name": "structural_incompleteness_grid",
 "version": "1.0.0",
 "d_theta": 5,
 "compression_rank": 3,
 "delta_values": [0.5, 1.0, 2.0],
 "sigma_values": [0.5, 1.0, 2.0],
 "samples": 10000,
 "repetitions": 50,
 "seed": 42,
 "loss": "quadratic",
 "compression": "linear",
 "shift_values": [0.0, 0.5],
 "confidence_level": 0.95,
```

```
  "output_file": "results_raw.json"
}
```

E.2 Regole di validità

1. compression_rank < d_theta
2. samples >= 1000
3. repetitions >= 10
4. confidence_level = 0.95 (fissato)
5. seed non modificabile post-preregistrazione

E.3 Blocco modifiche

Qualsiasi modifica a config.json invalida:

- risultati precedenti
- manifest SHA256

Nuova versione richiede nuovo tag.

APPENDICE F

MANIFEST SHA256

F.1 Obiettivo

Garantire integrità e immutabilità dei risultati.

F.2 Generazione hash

Su Linux / Mac:

sha256sum results_raw.json > manifest.sha256

Su Windows PowerShell:

Get-FileHash results_raw.json -Algorithm SHA256

F.3 Struttura manifest

File manifest.sha256:

a3f5b1e8c9... results_raw.json

F.4 Verifica indipendente

Revisore deve:

1. Rigenerare results_raw.json
2. Calcolare SHA256
3. Confrontare con manifest

Se hash coincide → replicazione identica.

APPENDICE G

REQUISITI HARDWARE E SOFTWARE

G.1 Software minimo

- Python 3.10+
- NumPy 1.23+
- Sistema operativo: Linux, macOS, Windows

Installazione:

pip install numpy

G.2 Hardware minimo

- CPU 2 core
- 4GB RAM
- Nessuna GPU richiesta

G.3 Scalabilità

Per d_theta > 50 o grid estesa:

- CPU 8 core
- 16GB RAM consigliati

G.4 Determinismo

Seed fissato garantisce replicabilità bit-level.

APPENDICE H

TEST AUTOMATICI

H.1 Test collasso

Verificare:

assert np.allclose(B @ theta1, B @ theta2)

H.2 Test regret positivo

assert mean_regret > 0

H.3 Test intervallo confidenza

assert ci_low > 0

H.4 Test consistenza bound teorico

Se strongly convex:

assert mean_regret >= (mu/8)*delta**2 – tolerance

H.5 Script test.py

python -m unittest test.py

Tutti test devono passare prima di generare manifest.

APPENDICE I

VERSIONAMENTO E AUDIT

I.1 Versionamento semantico

Formato:

MAJOR.MINOR.PATCH

- MAJOR → modifica teorica
- MINOR → estensione esperimenti
- PATCH → correzione tecnica

I.2 Tag Git

git tag v1.0.0

git push origin v1.0.0

I.3 Deposito archivio

Archiviare:

- codice
- config
- risultati

- manifest

In:

- Git repository
- Archivio persistente (es. Software Heritage)

I.4 Audit esterno

Revisore indipendente deve poter:

1. Clonare repo
2. Eseguire main.py
3. Verificare hash
4. Confermare bound

I.5 Criterio di integrità completa

Teoria + codice + manifest + versione taggata = pacchetto chiuso.

PARTE XI

STATUTO LEGALE ESTESO

11.1 TITOLARITÀ DELL'OPERA

L'intera opera, comprensiva di:

- Teoria matematica
- Formalizzazione logica
- Codice software
- Struttura algoritmica
- Documentazione tecnica
- Appendici
- Configurazioni JSON
- Script Python
- Manifest e procedure di audit

è creazione intellettuale originale dell'Autore.

Autore:
Pierernesto Di Paolo

Tutti i diritti sono riservati.

11.2 AMBITO DELLA PROTEZIONE

La protezione riguarda:

1. Testo dell'opera
2. Struttura formale dell'esposizione
3. Organizzazione modulare della teoria

4. Architettura dimostrativa
5. Codice originale
6. Configurazione sperimentale
7. Protocollo di preregistrazione

La protezione non riguarda:

- risultati matematici astratti in quanto tali (principio generale del diritto d'autore)
- fatti matematici isolati
- teoremi in forma astratta ri-espressi indipendentemente

11.3 RISERVA INTERNAZIONALE DEI DIRITTI

L'opera è protetta ai sensi:

- Convenzione di Berna
- Direttive UE in materia di diritto d'autore
- Normativa nazionale applicabile

Sono riservati:

- Diritti di riproduzione
- Diritti di distribuzione
- Diritti di modifica
- Diritti di traduzione
- Diritti di sfruttamento economico

- Diritti di adattamento
- Diritti di incorporazione in sistemi commerciali

11.4 CLAUSOLA DI NON UTILIZZO PER TRAINING ML

È espressamente vietato:

- utilizzare l'opera per addestramento di modelli di machine learning
- incorporarla in dataset di training
- usarla per fine-tuning di LLM
- sfruttarla per modelli proprietari

senza autorizzazione scritta dell'Autore.

11.5 CLAUSOLA DI NON DEPLOYMENT CLINICO

L'opera:

- non è un dispositivo medico
- non è approvata per uso clinico
- non è certificata per decisioni sanitarie reali

Qualsiasi implementazione clinica:

- richiede validazione regolatoria
- richiede trial formali
- richiede autorizzazione dell'Autore

11.6 CLAUSOLA DI CITAZIONE

È consentito:

- citare brevi estratti
- discutere risultati
- commentare la teoria

a condizione di:

1. attribuzione esplicita
2. citazione completa
3. nessuna modifica sostanziale del contenuto

11.7 DIVIETO DI MODIFICA NON AUTORIZZATA

È vietato:

- pubblicare versioni modificate
- produrre opere derivate sostanziali
- alterare dimostrazioni mantenendo attribuzione
- rimuovere clausole legali

senza consenso scritto dell'Autore.

11.8 VERSIONAMENTO UFFICIALE

Le versioni ufficiali sono esclusivamente:

- quelle rilasciate dall'Autore
- identificate da numero di versione
- accompagnate da manifest SHA256

Qualsiasi altra versione è da considerarsi non ufficiale.

11.9 RESPONSABILITÀ

L'opera è fornita:

"così com'è"

L'Autore non risponde di:

- uso improprio
- applicazioni non validate
- interpretazioni scorrette
- danni derivanti da implementazioni autonome

11.10 CLAUSOLA DI INTEGRITÀ TEORICA

La teoria non deve essere:

- reinterpretata come garanzia di performance universale
- usata come certificazione automatica

- dichiarata applicabile oltre le condizioni matematiche dimostrate

11.11 SEPARAZIONE TRA TEORIA E IMPLEMENTAZIONE

Il teorema matematico:

- è risultato teorico

Il software:

- è dimostrazione sperimentale

Non vi è equivalenza automatica tra:

dimostrazione teorica
e
affidabilità pratica universale.

11.12 PRIORITÀ SCIENTIFICA

La priorità scientifica è stabilita da:

- data di pubblicazione
- manifest crittografico
- eventuale deposito pubblico verificabile

11.13 NON-CLAIM DI ESCLUSIVA SUI FATTI MATEMATICI

L'Autore non rivendica monopolio su:

- concetti generali di minimax
- KL divergence
- TV
- strong convexity

La rivendicazione riguarda:

- struttura specifica dell'incompletezza decisionale sotto compressione
- organizzazione formale integrata
- architettura teorica complessiva

PARTE XII

LIMITI E NON-CLAIM

12.1 AMBITO DI VALIDITÀ MATEMATICA

Il Teorema di Incompletezza Decisionale sotto Compressione vale se e solo se sono soddisfatte le seguenti condizioni:

1. Esistenza di almeno due stati θ_1, θ_2 tali che:

$C\#P_{\theta_1} = C\#P_{\theta_2}$

2. Differenza decisionale:

$a^*(\theta_1) \neq a^*(\theta_2)$

3. Separazione stretta del rischio:

$\forall \theta, a \neq a^*(\theta)$:

$R(\theta,a) > R(\theta,a^*(\theta))$

4. Integrabilità della loss

5. Esistenza dell'integrale di rischio

Se una sola di queste condizioni viene meno, il teorema non è applicabile.

12.2 CASI IN CUI NON SI APPLICA

12.2.1 Compressione sufficiente

Se C è statisticamente sufficiente per il compito:

$\forall \theta_1, \theta_2$:

$C\#P_{\theta_1} = C\#P_{\theta_2}$

$\Rightarrow$

$a^*(\theta_1) = a^*(\theta_2)$

Allora:

Non esiste collasso decisionale.

Il regret può essere nullo.

12.2.2 Problemi con azione unica

Se $|A| = 1$:

Non esiste decisione alternativa.

Il teorema è trivialmente non applicabile.

12.2.3 Loss non separata

Se esistono $\theta, a \neq a^*(\theta)$ tali che:

$R(\theta,a) = R(\theta,a^*(\theta))$

Allora:

ε può essere zero.

Il bound positivo non è garantito.

12.2.4 Problemi degeneri

Se $P_{\theta_1} = P_{\theta_2}$ già nello spazio originale:

Non è la compressione a generare incompletezza.

Il problema è già non identificabile.

12.3 COSA IL TEOREMA NON DIMOSTRA

12.3.1 Non dimostra universalità clinica

Non dimostra che:

"Ogni sistema medico basato su medie è errato."

Dimostra solo:

Se esiste collasso decisionale → regret positivo.

12.3.2 Non dimostra superiorità di qualsiasi alternativa

Non dimostra che:

"Un sistema non compresso è automaticamente migliore."

Dimostra solo che:

La perdita strutturale genera limite inferiore.

12.3.3 Non elimina il ruolo della statistica

Non nega:

- Neyman–Pearson
- Bayes
- minimax classico
- sufficienza statistica

Li integra in un quadro decisionale strutturale.

12.3.4 Non è teorema di Gödel

Non riguarda:

- incompletezza logica formale
- sistemi assiomatici autoreferenziali

È incompletezza informazionale-decisionale.

12.4 CONDIZIONI NECESSARIE ESPLICITE

Il collasso decisionale richiede:

1. Differenza reale tra superfici di rischio
2. Esistenza di direzione invisibile:

$\exists v$ tale che:

$DC(\theta)[v] = 0$

ma

$D\ a^*(\theta)[v] \neq 0$

3. Non degenerazione locale del rischio

Se DC ha rango pieno rispetto alla dimensione decisionale:

Il collasso differenziale non può verificarsi.

12.5 CASI LIMITE

12.5.1 Strong convexity assente

Se il rischio non è convesso:

Regret positivo può esistere ma bound quadratico non garantito.

12.5.2 Parametro continuo non compatto

Se Θ non compatto:

Il minimo ε globale può non esistere.

Serve compattezza o argomento locale.

12.5.3 Loss non limitata

Se loss non limitata:

Bound informazionali richiedono controllo momenti.

12.6 CONDIZIONI PER POSSIBILE FALSIFICAZIONE

La teoria è falsificabile se si verifica:

Esistono θ_1,θ_2 tali che:

$C\#P_{\theta_1} = C\#P_{\theta_2}$

$a^*(\theta_1) \neq a^*(\theta_2)$

ma

$\exists \pi$ tale che:

$\text{Regret}(\theta_1,\pi) = 0$

$\text{Regret}(\theta_2,\pi) = 0$

Questo contraddirebbe la separazione stretta.

12.7 LIMITI COMPUTAZIONALI

La teoria:

- non garantisce computabilità efficiente
- non fornisce algoritmo universale ottimo
- non risolve problemi NP-hard

È risultato strutturale, non algoritmo universale.

12.8 DISTINZIONE TRA LIMITE TEORICO E PERFORMANCE EMPIRICA

Un sistema può:

- avere limite strutturale > 0
- ma in pratica errore molto piccolo

La teoria riguarda limite inferiore, non grandezza assoluta empirica.

12.9 AMBITO FILOSOFICO LIMITATO

Non è:

- teoria ontologica generale
- teoria metafisica

- dichiarazione universale sulla conoscenza

È teorema matematico su compressione decisionale.

12.10 CONCLUSIONE PARTE XII

La teoria:

- è forte ma circoscritta
- è strutturale ma non universale
- è matematica ma non retorica
- è falsificabile
- è compatibile con la statistica classica

Questo rende l'opera:

- difendibile
- non dogmatica
- epistemicamente matura

PARTE XIII

IMPLICAZIONI EPISTEMOLOGICHE

13.1 RAPPRESENTAZIONE VS DECISIONE

La statistica classica è costruita su:

- stima
- sufficienza
- inferenza
- rappresentazione del parametro

Il presente lavoro mostra una distinzione formale:

Sufficienza statistica ≠ sufficienza decisionale.

13.1.1 Oggetto primario

Nell'inferenza classica l'oggetto è:

θ

Nel quadro decisionale strutturale l'oggetto primario è:

la superficie di rischio $R(\theta,a)$

La decisione non dipende solo dalla rappresentazione di θ, ma dalla struttura dell'argmin su a.

13.2 COMPRESSIONE COME OPERATORE STRUTTURALE

La compressione C non è solo riduzione dimensionale.

È operatore:

$T_C : L2(P) \rightarrow L2(P)$

che proietta su sottospazio visibile.

Questa proiezione:

- elimina componenti ortogonali
- riduce divergenze
- collassa superfici di rischio

La compressione è dunque operatore strutturale sullo spazio delle decisioni.

13.3 CATEGORIA DEL LIMITE STRUTTURALE

Il risultato centrale mostra:

Esistono limiti decisionali generati non da rumore, ma da struttura informazionale.

Non è un limite statistico nel senso di:

$n \to \infty$

È un limite strutturale permanente.

13.4 DIFFERENZA TRA ERRORE E INCOMPLETEZZA

Errore classico:

deriva da campionamento finito.

Incompletezza strutturale:

persiste anche con campione infinito.

Formalmente:

$\lim_{n\to\infty} \text{Regret}_n \geq \varepsilon > 0$

Questo è il salto concettuale.

13.5 DECISIONE COME OGGETTO GEOMETRICO

La superficie:

$\theta \mapsto a^*(\theta)$

definisce varietà decisionale.

Il collasso corrisponde a:

identificazione forzata di punti distinti sotto mappa C.

È fenomeno geometrico prima che statistico.

13.6 SUFFICIENZA RIFORMULATA

Sufficienza statistica classica:

C è sufficiente se conserva likelihood ratio.

Sufficienza decisionale:

C è sufficiente se preserva $\text{argmin}_a R(\theta,a)$.

Questa distinzione non è solo terminologica.
È strutturale.

13.7 RELAZIONE CON TEORIA DELL'INFORMAZIONE

Data processing inequality implica:

Divergenza_Z ≤ Divergenza_X

Ma il risultato mostra che la perdita di divergenza può tradursi in perdita irreversibile di ottimalità.

Il legame è formale, non analogico.

13.8 RAPPRESENTAZIONE NON È NEUTRA

Ogni scelta di compressione:

- seleziona sottospazio
- impone topologia decisionale
- determina possibilità o impossibilità di distinguere stati

La rappresentazione è decisione implicita.

13.9 LIMITI DEL RIDUZIONISMO

Se la riduzione elimina direzioni decisive, nessun incremento di dati compensa.

Il limite è intrinseco alla struttura del modello.

13.10 IMPLICAZIONI PER SISTEMI AUTOMATICI

Ogni sistema automatico che:

- opera su feature compresse
- ignora direzioni latenti rilevanti

è soggetto a limite inferiore teorico.

Questo non implica inutilità.
Implica limite strutturale misurabile.

13.11 COMPATIBILITÀ CON LA SCIENZA CLASSICA

La teoria:

- non sostituisce Bayes
- non sostituisce minimax
- non sostituisce sufficienza

Introduce una categoria aggiuntiva:

Incompletezza decisionale da compressione.

13.12 CONSEGUENZA META-TEORICA

La decisione non è funzione solo della distribuzione.
È funzione della rappresentazione della distribuzione.

La scelta di rappresentazione è atto strutturale.

13.13 POSIZIONAMENTO EPISTEMICO

Il risultato è:

- matematico
- verificabile
- non metaforico
- non ideologico
- non retorico

È enunciato strutturale su limiti decisionali.

13.14 CONCLUSIONE FINALE DELL'OPERA

Abbiamo dimostrato:

1. Esistenza di incompletezza decisionale sotto compressione
2. Bound positivo minimax
3. Quantificazione informazionale
4. Versione differenziale locale
5. Formalizzazione machine-checked blueprint
6. Implementazione replicabile
7. Protocollo di audit
8. Delimitazione dei limiti

L’opera è ora:

- matematicamente chiusa
- formalmente coerente
- empiricamente verificabile
- logicamente delimitata

CHIUSURA

L'opera completa comprende:

Parte I - Fondazione concettuale

Parte II - Teorema minimax generale

Parte III - Derivazioni informazionali

Parte IV - Geometria differenziale

Parte V - Algebra funzionale

Parte VI - Estensioni generali

Parte VII - Regret strutturale avanzato

Parte VIII - Formalizzazione machine-checked

Parte IX - Programma empirico replicabile

Parte X - Appendici tecniche complete

Parte XI - Statuto legale

Parte XII - Limiti e non-claim

Parte XIII - Implicazioni epistemologiche